JPRS 84413

27 September 1983

USSR Report

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

No. 74

FBIS FOREIGN BROADCAST INFORMATION SERVICE

NOTE

JPRS publications contain information primarily from foreign newspapers, periodicals and books, but also from news agency transmissions and broadcasts. Materials from foreign-language sources are translated; those from English-language sources are transcribed or reprinted, with the original phrasing and other characteristics retained.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by JPRS. Processing indicators such as [Text] or [Excerpt] in the first line of each item, or following the last line of a brief, indicate how the original information was processed. Where no processing indicator is given, the information was summarized or extracted.

Unfamiliar names rendered phonetically or transliterated are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear in the original but have been supplied as appropriate in context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by source.

The contents of this publication in no way represent the policies, views or attitudes of the U.S. Government.

PROCUREMENT OF PUBLICATIONS

JPRS publications may be ordered from the National Technical Information Service (NTIS), Springfield, Virginia 22161. In ordering, it is recommended that the JPRS number, title, date and author, if applicable, of publication be cited.

Current JPRS publications are announced in Government Reports Announcements issued semimonthly by the NTIS, and are listed in the Monthly Catalog of U.S. Government Publications issued by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

Correspondence pertaining to matters other than procurement may be addressed to Joint Publications Research Service, 1000 North Glebe Road, Arlington, Virginia 22201.

JPRS 84413

27 September 1983

USSR REPORT

CYBERNETICS, COMPUTERS AND AUTOMATION TECHNOLOGY

No. 74

CONTENTS

HARDWARE

HARDWARE

SM-4 MINICOMPUTER PRODUCTION BEGUN

Kiev TASS in English 1245 GMT 7 Jul 83

[Excerpt] The Kiev Elektronmash (Electronic Machine-Building) Amalgamation has started the production of fourth-generation computers designed for the automation of production processes. By its price and dimensions the new computer -- the SM-1420 controlling machine -- belongs to the series of the so-called smaller computers developed within the framework of the single program of CMEA countries. The previous model of this series, the SM-4, has been produced by the Amalgamation for three years and is in high demand both in the Soviet Union and abroad.

Thanks to the utilization of the most up-to-date element base, the new computer, which preserved all the strong points of its predecessor, has twice as great efficiency and about ten times as great memory volume.

CSO;1852/11 - E

DEVELOPMENT OF NEW YES-1061 COMPUTER DISCUSSED

Minsk SOVETSKAYA BELORUSSIYA in Russian 3 Apr 83 p 1

[Interview with Yuriy Vladimirovich Karpilovich, chief engineer at the Minsk Computer Production Association, by A. Sarapkin and A. Safronenko; date and place not given: "YeS-1061 Is on Good Path"]

> [Text] The Minsk Computer Production Association is among the enterprises that constantly pursue the course of developing modern machines capable of affording an increase in labor productivity without increasing personnel. The state commission recently authorized production of the new modernized YeS-1061 computer developed at this enterprise. Yu. V. Karpilovich, chief engineer at the association, chief designer of the new computer and laureate of the USSR State Prize, answers the questions of our correspondents.

[Question] Yuriy Vladimirovich, how does the YeS-1061 computer compare to the other Minsk computers?

[Answer] As is known, in the Basic Directions for USSR Economic and Social Development for the Years 1981-1985 and for the Period to 1990, the task of raising the technical standard of computers was set. Since about the start of the current five-year plan, we have been engaged in designing the YeS-1061 and preparing it for production. We did not start from zero. The predecessor was the YeS-1060; like the current, latest modification, this computer too was prepared for series production together with the Moscow Scientific Research Center for Electronic Computer Technology (NITsEVT). It seems proper to compare it to this computer. First, the YeS-1061 outputs the required information twice as fast. Also, it requires less labor in manufacture, is more amenable to manufacturing processes and takes up less production space because it is significantly more compact than its predecessor. This was achieved because one-third fewer standard modular system cards (TEZ's) were used in the new computer.

The mechanical design of the computer has also become simpler and hardly any of the original complex parts were added. At the same time, many old ones were eliminated. It has been standardized with other Minsk computers which allows preserving the continuity of the Unified System. We have reduced the hardware for the processor, the computer brain as we refer to it. Development of this new model was attended by filing of a number of applications for invention. In cost, the YeS-1061 will

not be more expensive than its predecessor. The number of servicing personnel will also not increase. Power consumption will be reduced 20 percent, and metal consumption, 15 percent. Much thought went into raising computer reliability. The State Commission concluded we had pretty good results. YeS-1061 reliability is 1.5-fold higher. This was achieved through reduction in hardware and development of new, expedient and efficient facilities for fault diagnostics.

The annual economic effect from replacing one old model with the new computer is over a million rubles. The State Commission believes it meets the world standards and current requirements.

[Question] How are things going for certification of the new computer for the State Mark of Quality?

[Answer] The commission recognized that the YeS-1061 technical level fully meets the highest category of quality. But it is still too early to submit it for certification. The computer has to be well proven by the user. We expect that when mass production begins, that is next year, the new computer will be awarded the highest certification. It must be said that manufacturing high quality products has become a tradition for our collective. The YeS-1035 and YeS-1060 computers now produced by the association bear the distinguished pentahedron. Much of this is due to the movement under the motto "High Quality of Development and Manufacture for Each Product in the Alliance between Science and Industry." Association specialists and employees actively participate in it.

[Question] Could you give us more details on the cooperation between employees in the Moscow Scientific Research Center for Electronic Computer Technology and the Minsk association?

[Answer] One may say that we have solidified a new form of creative cooperation with the collective at the scientific institution. Earlier, such problems were resolved, as a rule, only through the efforts of specialists in the scientific research institute. But then the decision was made to establish a special department in the association; this department was formed from representatives of the computer debugging shop and the plant design bureau. Employees from the scientific research center also participate in the stage of drafting the specifications and logic development of the processor structure. Mutual interest in the final result, the search for the best solutions, enthusiasm, and the fusion of mature experience with the fervor of youth all make a fine contribution to the task. Making prototypes has become a major affair for both collectives. Much ideological, political and educational work has facilitated this. The tone in the competition has been set by the communists and Komsomol members. The activity of the specialists is constantly the center of attention from the party committees at the research center and the association. The most critical problems are repeatedly discussed in joint meetings by the party committees and unity of actions is arrived at. Starting with the working design, the main share of concerns is assumed by the industrial workers. I am gratified to name those who made the greatest contribution to the common matter. The ones from Moscow are V. V. Przhiyalkovskiy, general designer of the Unified System of Computers (he is, incidentally, an alumnus of our collective), Yu. A. Kokhanov, I. S. Khramtsov, S. K. Ivanov and others who are highly skilled specialists at the research center. Outstanding contributors at the association are V. P. Shershen', A. N. Vitalisov, B. F. Shadrin, N. Z. Pozdnyakov, D. A. Rzheutskiy, V. A. Fedorovich, V. A. Chernitskiy and V. V. Velikan.

It is simply not possible to name everyone who ensured success. I would like to address special words of gratitude to the young people at the plant. The young Komsomol collective including designers and developers from the debugging section actively participated in developing and adjusting the new computer. Boldness of thought and the aspiration to bring the future closer are qualities of A. S. Grigor'yev, S. A. Kozell, G. K. Davidovich, L. P. Vasilevskiy, G. A. Vol'skiy, S. V. Klochkov, L. A. Kononov, A. G. Rodionov and others.

[Question] As is known, industrial workers like thoroughly developed prototypes and make various changes unwillingly. What difficulties did you have to overcome in this respect?

[Answer] Developing a prototype is of course no simple matter. Digressions from manufacture of the basic product of the best specialists are unavoidable. Required here are flexibility, high efficiency and at times the selflessness of all participants in the development. All this is based, I would say, on one thing: an active civic attitude and the understanding by the people of the importance of what is being developed to the state. Then one can surmount any obstacles. To a considerable extent thanks to this, we saved a year as the minimum in developing the YeS-1061 computer. As is known, in our work, time plays a great role. And another important point. When industrial workers become co-authors of a product, they work very efficiently and grow as specialists. In this is the pledge that the YeS-1061 will be improved even further. We intend to significantly improve the technical characteristics of the present prototype, to make the computer less expensive, to standardize some of its parts and raise the reliability. In the process of developing the prototype, we also thoroughly prepared the technology and are now in a position to start series production of the computer already.

[Question] In your view, how do you see the future of the YeS-1061 computer?

[Answer] I think, one may say, that it will be one of the main products of the association to the end of the five-year plan. For a computer, such a period of series production is a very long life. I am confident that it will be exported. Of course, we have quite a bit of work to do. Perhaps the main thing here is to establish efficient coordination with the numerous subcontractors, bring production planning to a new level and constantly improve the technology. We also have to further increase responsibility, executive discipline and the state of organization in all stages of production. Such is the requirement of life and the decisions of the November (1982) Plenum of the CPSU Central Committee.

[Question] Yuriy Vladimirovich, thank you for the interview. We wish you further success in developing modern computers.

[Answer] Thank you.

8545
CSO: 1863/168

YES-6903 RANDOM NUMBER INPUT UNIT

Moscow EKONOMICHESKOYE SOTRUDNICHESTVO STRAN-CHLENOV SEV in Russian No 1, 1983 p 79

[Product announcement]

[Text] Unit is designed for generation and input of random numbers to a computer.

It is recommended for use in solving problems on a computer by the Monte Carlo method (method of statistical tests). Possible fields of application: solving problems of mathematical physics, solving differential equations in partial derivatives, calculation of continual integrals, modeling of queueing systems, modeling of communication systems, solving reliability theory problems, solving problems in nuclear physics, modeling of game situations, modeling of economic, biological, ecological, meteorological and other situations.

Unit is made as a device in the Unified Series of Computers and connects to the multiplexor or selector channel. It is housed in a standard Unified System rack.

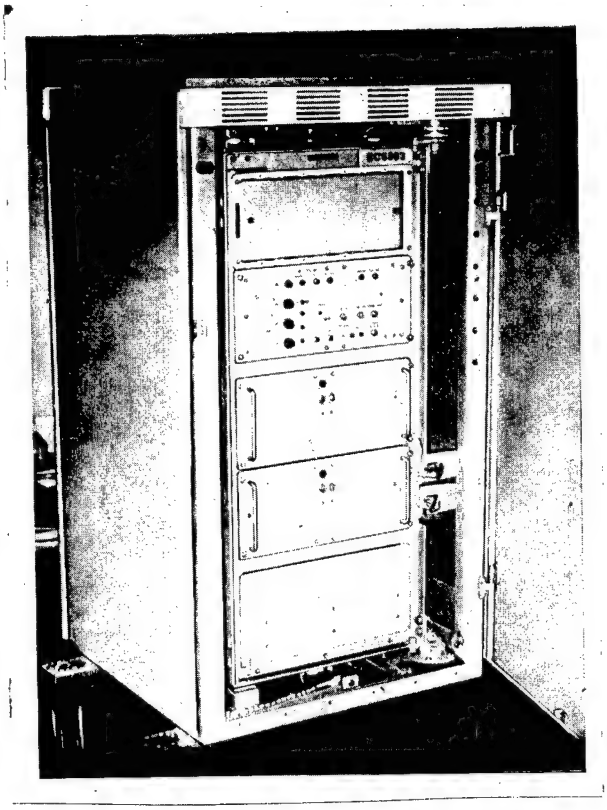
Device operation is based on using hardware implementation for generating random numbers with the necessary laws of distribution.

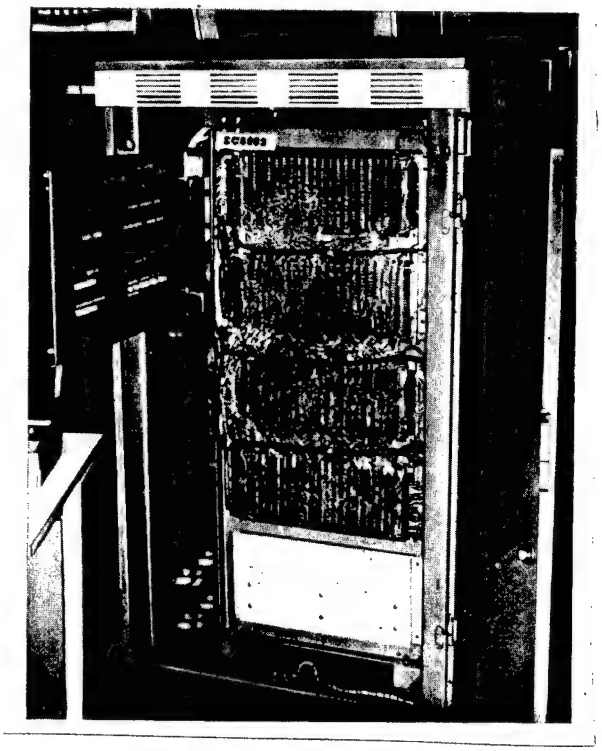
Compared to traditional software methods, using the YeS-6903 considerably increases the speed of random number generation, yielding thereby a five- to ten-fold savings in machine time on the average.

A complex of programs supporting standard access to the unit has been developed.

Basic Technical Characteristics

Operating mode	Random number generation; Pseudorandom number generation
Form of number representation	32-bit word with floating point
Maximum data transfer rate	800K bytes/second
Types of laws of distribution	Equipartition; Normal; Exponential; Any specified analytically by table or histogram
Power consumption	no more than 500 W





Weight	No more than 200 kg
Dimensions	650 x 700 x 1200 mm

Technical solutions are patentable.

Send all inquiries to: USSR, Kazan', 420084, ulitsa [street] Karl Marx, 10th department of computers.

[Photos on previous page show front view (top) and rear view (bottom)]

8545
CSO: 1863/168

UDC 681.142.4

MULTIPROCESSOR SYSTEMS: BASIC ORGANIZATION PRINCIPLES (REVIEW)

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 pp 3-10

[Article by Aleksandr Stepanovich Balabanov, "Multiprocessor Systems. Basic Organization Principles (Review)", in the section "General Questions of Control System Construction"]

[Text] This article is a brief review of the basic construction principles of multiprocessor computer systems (MPS), mainly general purpose ones, as well as some other parallel systems. Organizationof operation, architecture and structure of systems, and their features and advantages are considered. The interrelation is shown between the nature of the problems solved, system organization and the structure of its equipment. These questions are dealt with in greater detail in [1-13].

Generation and Development of Multiprocessor Systems. The most significant tendency in recent years in the creation of new digital computer structures and organization is the turn toward parallel and multiprocessor systems. The reasons for this are:

the need to raise digital computer performance for many applications;

exhaustion of the speed reserve of electronic circuits in known technologies;

the appearance on the market of inexpensive LSI circuits and microprocessors; and

the readiness of the ideological base for building MPS's (theoretical results and engineering designs).

The idea of joining several processors into a single system arose long ago [1, 6, 9-12]. In creating multimachine systems and those with reserving and replication, problems of reconfiguration, memory collectivization, switching, etc., were solved. Many architecture and technical solutions for the MPS's were found while improving general purpose single-processor digital computers. The need to raise the processor load and join input-output with central processing resulted in a multiprogram operating mode, and the appearance of an independent i/o channel and commutator at the main memory input. Multiprogramming required creation of complex operating systems. Expanding the functions of the i/o channel, and attempting to bring the satellite digital computer close to the host one [1] led to the

appearance in the machine of an advanced peripheral processor with a wide set of instructions. Combining operations in the central part of the digital computer required decentralization of the control device, isolation of relatively independent functional devices, the use of memory interleaving, and so forth (for example, in the GAMMA-60 and IBM STRETCH, in confluent machines [14]). To raise the load of the set of functional devices, the idea appeared of using them to execute several instruction flows at once [15,16]. A system with a high level of the internal language and hardware realization of system functions should be made as an asymmetric multiprocessor system (the experience of developing SYMBOL).

The following basic directions can be singled out in development of parallel systems:

development of systems with high reliability and survivability for essential and military control systems, on-board digital computers, etc.;

creation of super-performance systems, designed for a certain class of problems (PEPE, ILLIAC-IV, CDC STAR-100, OMEN-60);

creation of general purpose, powerful MPS's (systems of the firms Honeywell, Sperry Rand, IBM, Burroughs); and

construction of systems from a large number of microprocessors and other LSI circuits.

The Class of Systems Examined. Generally, MPS's include systems [1, 2, 6, 8, 17, 18] that have the following basic features: two or more flows of instructions and data; common memory; and integrated (uniform) operating system. These formed long ago, and are not so unique with respect to certain new systems that are close in their properties to conventional MPS's. The concept of an instruction stream is of little use in describing systems with an unconventional internal language, such as flow ones (data flow system) or systems with a high-level internal language. Complication of the structure erases the clear distinction between a complex processor and a system of simple processors. A tendency toward decentralization of certain functions of the operating system and complication of the internal memory structure has even been observed.

The most important characteristic of a parallel system is the range of realized parallelism. This range is rather wide in the MPS, providing universality of the system; however, MPS's may not be as good as other systems in terms of the effectiveness of realizing individual types of parallelism. To realize parallelisms (especially joint use of resources), parts of the system must interact with each other. The system's capabilities for realizing parallelisms can thus be roughly estimated by using the following two indices: universality of interactions (their functional capabilities), and interaction speed (their intensity). The classes of parallel systems shown in Fig. 1 can be distinguished by these two indices.

In loosely connected systems, the interaction speed is considerably lower than in the MPS; therefore, only parallelism of very large operation fragments is realized in them (in fact, if the system spends an amount of time on organizing and synchronizing parallel operation fragments comparable to the time for executing these fragments, then parallel execution can be slower than sequential). In problem-oriented

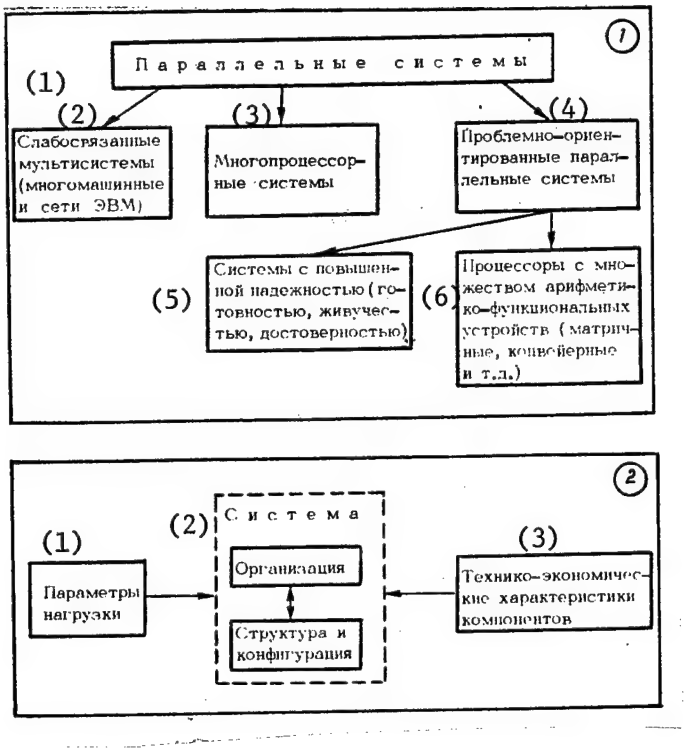


Fig. 1. Classes of Parallel Systems

Key:

1. Parallel systems
2. Loosely connected multisystems (multimachine and network computers)
3. Multiprocessor systems
4. Problem-oriented parallel systems
5. Systems with high reliability (availability, survivability)
6. Processors with a set of arithmetic-functional devices (matrix, pipeline, etc.)

Fig. 2. Characteristics of System Components and Load Characteristics

Key:

1. Load parameters
2. System; organization; structure and configuration
3. Technical-economic characteristics of components

parallel systems, interaction speed can be even higher than in an MPS, but the universality of interactions is much lower. Such systems thus realize special and degenerate types of parallelism, and are special-purpose ones [19]. For example, in systems with a high reliability a typical interaction is comparison of results, while in a matrix processor it is synchronous exchange of operands. Naturally, an actual system is frequently a combination of various classes of parallel systems. The classification of parallel systems is considered in [2, 18].

Types of Parallelism and Their Realization. The following basic types of parallelism can be singled out [3, 4, 20]:

1. Interproblem parallelism. Independent problems use common central resources (memory space, hardware, operating system programs, etc.).

2. Long parallel branches of an algorithm (subtasks).

3. Fast-branching algorithms. This type of parallelism, in contrast to the preceding one, is characterized by shorter and/or numerous branches of the algorithm and/or stronger dependence of the number of branches on the data.

4. Parallelism of related operations (local).

5. Synchronous parallelism of similar operators on data arrays using integrated operations and regrouping of data.

5a. Vector operations (a special case of 5).

Each type of parallelism requires special realization tools. Types 1 and 2 (and perhaps 3) are realized by a set of processors using an operating system (OS). The OS tools organizing the parallel execution are: for type 1 parallelism, a job planner and (preferably) a dispatcher; for type 2 parallelism, a dispatcher. Realization of type 3 parallelism requires using a direct access memory common to the processors, as well as faster control (dispatcher) functions than in conventional operating systems (for instance, by decentralization and hardware realization of functions).

Types 4 and 5 parallelisms are realized inside a processor with a set of arithmetic devices and a common control device. Two methods of realizing type 4 are known: in a confluent processor with look-ahead inspection of the program [14,21] and in a processor with a vector instruction [3,4]. Types 5 and 5a are realized in matrix and pipeline processors, respectively [19]. Systems with such processors are problem-oriented.

Most user tasks have different types of parallelism in sequence, or even at the same time. System tasks also have types 1, 2 and 3 parallelisms (type 3 is characteristic for certain OS nucleus functions). It is thus preferable to use different types of parallelism in one system [3,12,5]. There are various approaches to this: dynamic restructuring (adaptation) of the structure [12, 22-25]; multicomputer operation of special-purpose and highly organized processors [24-28]; and construction of combined structures [3,4]. The most universal means of realizing heterogeneous parallelism will be systems with a flow organization (data flow). Parallelism of types 3 and 4 is realized well by one flow processor using the same mechanism [29,30], while that of types 1 and 2 is realized by a set of such processors [31]. However, the performance and efficiency of a flow system drops sharply for problems with a low degree of parallelism.

Below, we shall deal primarily with systems realizing types 1, 2 and 3 parallelism.

Types of MPS Functional Organization. The MPS structure and organization must flow logically from the characteristics of the system components and the load characteristics; in particular, of the types of parallelism (Fig. 2). Correspondingly, [25] singles out two MPS classes for the two types of components (fast MIC circuits

and general purpose LSI circuits with microprogramming): of complex processors with a common memory; and of a set of small microprocessors with a local memory.

MPS organization can be viewed from the standpoint of distribution of functions among processors, using such characteristics as division of functions into classes, representation of classes of functions for the processors responsible for them, the degree and selectivity of assigning functions to processors, centralization of functions, etc. For a processor to be able to execute certain functions, it must have access to the appropriate resources (hardware, software, data). We shall therefore describe the organization of systems using a functional structure of central resources (FSTsR), represented graphically. To do this, we single out four basic classes of resources: R, the command processor (including instruction sequence control, register memory and others); A, arithmetical (functional) resources for application problems; P, main memory; and U, system control. In the U class, it is useful to single out thefollowing subclasses (levels) of control: dispatcher of central resources; job planner; physical i/o control, and others.

The functional structure of central resources is the graph on which the connection of the processor (represented as a square) with the resource shows that the given resource is interactively accessible to the processor. The possibility of parallel use of the resource is shown by duplicating the resource representation on the FSTsR graph. Below, the FSTsR is used to list and display (Fig. 3) some actual (or logical) types of MPS organization (the key to their names is the system control organization).

Conventional MPS With Complete Centralization of Control (Fig. 3,b). The functions of system control can "flow" from one processor to another, but cannot be performed by two processors at the same time. All the processors in the system are functionally identical, since they have complete access to all resources.

Conventional MPS With Decomposition of Control (Fig. 3,b). This type of system is close to the preceding one (all the processors have complete symmetric access to all resources). It differs in that the system control is decomposed (a special case of decentralization); i.e., several processors can execute the control functions of different subclasses at once (U_1, U_2, U_3).

"Master-Slave" Type MPS (Fig. 3c). System tasks are separate from application ones, centralized and rigidly assigned to the master processor.

MPS With Decomposed (Distributed) Master Processor (Fig. 3,e). In this type of system (as in the preceding one), the application and control tasks are executed on different processors. The special feature is that different subclasses of the control resources are allocated (U_1, U_2, U_3) and each of them is rigidly assigned to one processor. An example of such an MPS is the SYMBOL system.

Hybrid (Combined) Types of Organization. The features and principles of the above organization types can be combined by forming hybrid types of systems. An example is the "El'brus" system (Fig. 3, e). The basic part of the OS tools (U_1) is realized in it the same as in the conventional MPS, but the functions of physical i/o control (U_2) are assigned to a special (peripheral) processor [27].

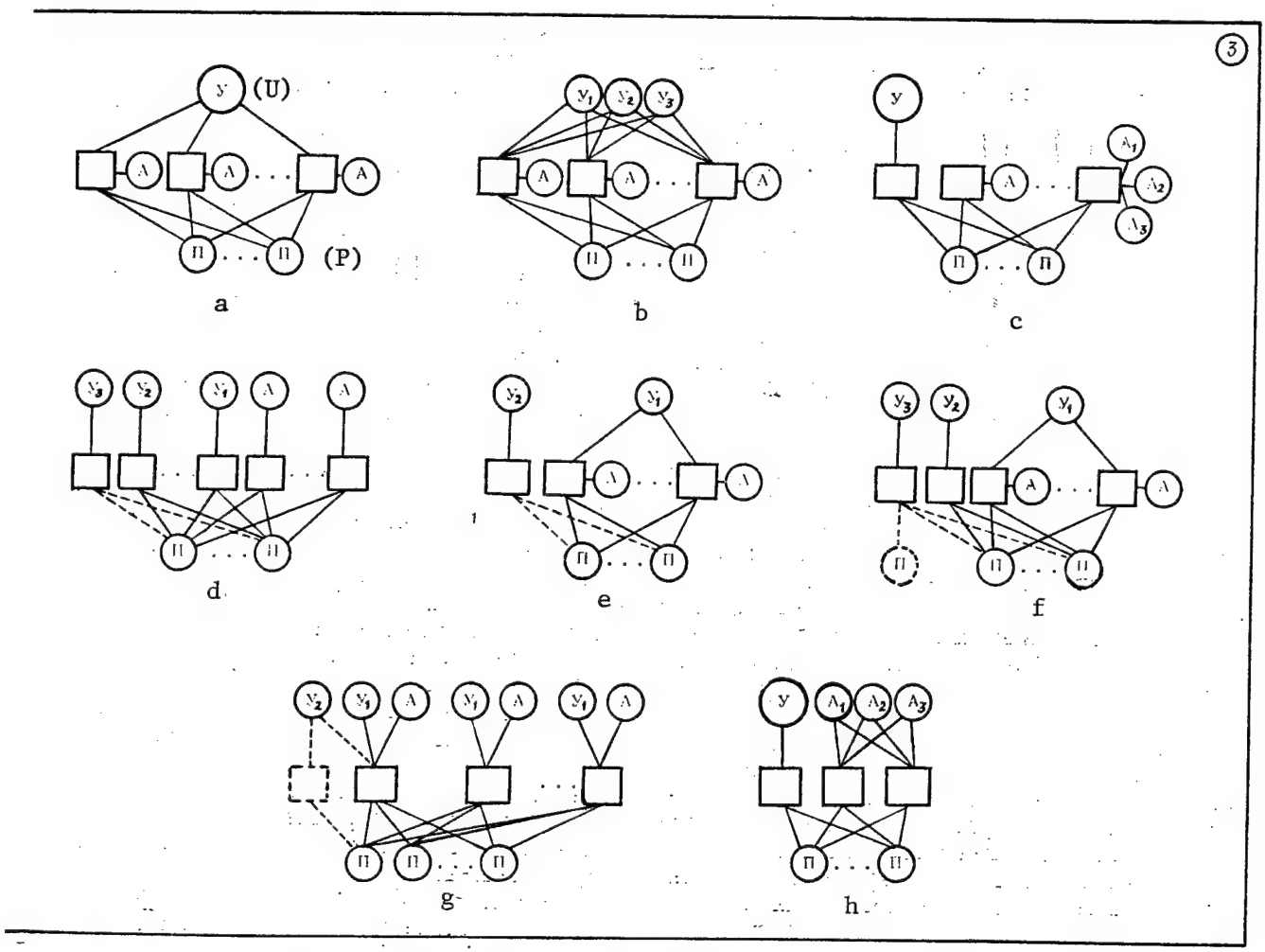


Fig. 3. Types of MPS Organization

An example of another hybrid type of MPS are certain systems of the CDC company (Fig. 3,f). Certain dispatcher control resources (U_1) in these systems are accessible to the processors performing application tasks; most of the control resources (U_2) are rigidly assigned to a control processor, while the functions of physical i/o control (U_3) are assigned to a peripheral processor [1]. The flow system described in [31] has a similar FSTsR, but the semantics of its resources differ somewhat.

MPS With Decentralization of Dispatcher Functions (Fig. 3, g). Each processor has access to its own copy of the control resource of the same subclass U_1 (supervisor, dispatcher). Some of the control resources U_2 (job planner and others) are also centralized, and can be assigned to a certain processor. Such organization will probably characterize the CHOPP system being designed (Columbia Homogeneous Parallel Processor).

We should emphasize that in the given system each copy of the dispatcher can control all the central resources of the system, or at least directly affect the operation of another dispatcher. Decentralization of control is thus attained without loss of system connectivity. A contrasting example of another decentralization is use of the OS "with separate job execution in each processor" [1], where the system is functionally divided into independent subsystems, even if the equipment structure is symmetric. Such a system is close in its properties to a multimachine one.

The system whose FSTsR is shown in Fig. 3,h differs in that the arithmetic devices (A_1, A_2, A_3) are collective resources for a group of processors. Control in this system is organized by the "master-slaves" principle.

To describe the organization of certain other MPS's, it is useful to single out additional subclasses of resources. For instance, control data and control programs (circuits) can be differentiated.

The above types of organization are single-level. For certain new systems (notably microprocessor ones), a hierarchical organization with distributed memory is characteristic. In other systems, readjustment of the equipment is used. Several degrees of operation of access to the resources can be used to clearly describe such system properties when constructing the functional structure of resources.

MPS Organization and Structure. There are interconnections and dependences between the various characteristics of functional organization, configuration, structure and technical parameters of system equipment. A complete investigation of these issues goes beyond the scope of this article. Above are indicated the dependences among types of parallelism of problems, types of components and characteristics of the system. We shall indicate certain other dependences using the FSTsR.

Based on the functional organization, an idea can be obtained of the operating modes, capabilities of each processor, types of interactions among processors, topology of their connections, coordination, etc. For example, the hardware-realized part of resource A and U, accessible to the processor, uniquely determines the processor instruction system.

Structure of Connections in the Equipment. If the processor has access to the functions of physical i/o control, then it is connected with the i/o channels. If it has access to the functions of the supervisor and dispatcher of the central resources and tasks, it must have connections with other processors and access to a greater part of the main memory. In general, symmetry of functional structure entails symmetry of system equipment (Fig. 3,a).

Method of Realizing the Control Resource. Usually, system control functions are realized mainly by programs, due to their complexity. If control is centralized and accessible to all processors (Fig. 3,a) then it is best to store OS programs in the common main memory.

If a narrow subclass of control resources is assigned to one processor (with a very large total number of processors in the system), then it is not advisable to realize this resource by the hardware; i.e., by the built-in method (Fig. 3,e).

Dispatcher functions should be decentralized if there is a very large number of processors in the system (Fig. 3,g).

Polarization of Control and Arithmetic Resources. If a considerable portion of the system's control resources are concentrated in one processor (Fig. 3,c) and the load is great (for example, short user tasks predominate or the number of system processors is large), then this processor should be freed from the arithmetic resources (Fig. 3,c,d,e,f). If a certain processor is freed from the control functions (Fig. 3,c,d), then it should be equipped with more powerful arithmetic resources; for instance, several arithmetic devices should be connected to it (Fig. 3,c).

The more powerful the arithmetic resources used, the more important it is to ensure their high load. This requires ensuring access of the processors to a large part of the main memory where the active tasks are stored.

Access to Main Memory. Collectivization of the main memory is dictated by the need to balance the load (transmission of tasks from one processor to another), exchange information, save memory, store centralized system data and a large part of the OS routines.

Providing symmetric processing mode in the system [1] requires that each processor have complete access to all central resources (Fig. 3,a); in particular, direct access to the entire main memory (which is only possible given sufficient address tools of the processors and an efficient "processors--memory" commutator).

If the processor is specialized in such functions as job planning, periphery control, etc., then this processor's access to the main memory can be limited (in the sense of the type and speed of references, size of the memory portion accessible), which is shown by the dotted line in Fig. 3,d,e,f. In general, a local processor memory should be used with specialization in a narrow subclass of system resources.

Conventional MPS. It follows from the structure of the resources (Fig. 3,a) that the conventional MPS is symmetric. In other words, all the processors have a common instruction system, and direct, uniform access to the common main memory and to all the periphery [1]. System tables and queues are centralized, as in the one-processor system. The OS programs should be stored in the main memory as reenterable modules (in one device each). Such systems are very flexible, since each processor can itself fulfill any control functions as needed, without interfering with other processors. Optimal distribution of external interrupts among processors is also possible [32]. Such a control flexibility is especially attractive for the time-sharing mode, since it enables the system to dynamically adapt to an unstable stream of enquiries and tasks. Symmetry imparts to such MPS's a high survivability relative to processor failures.

A complex, fast commutator is required to ensure direct, complete access of the processors to the main memory. The cost of such a commutator with a conventional structure rises rapidly, and becomes excessive with 10-20 processors [1,20].

Ordered access to variable system data is required to ensure correctness of resource and task control. This is done most easily if two processors are prohibited

from simultaneously going into the privileged mode (Fig. 3,a). This enables use of the operating system of a single-processor digital computer in the MPS at minimal extra cost. However, control in this case becomes the "bottleneck" of system performance. Considering that different control tasks operate basically with different parts of the system data, simultaneous execution of these tasks can be organized by using synchronization (deadlock) of tasks at critical sections, where conflicting accesses to the data are possible (Fig. 3,b). Privileged intra-processor functions can be similarly decentralized [1]. The execution time of critical parts of the control tasks can be shortened by their circuit realization (operations on queues, memory control [26,27]). Partial decentralization of the structure of system tables and queues, described in [33], enables several processors to simultaneously execute part of the dispatcher functions (though this is already departing from the conventional organization).

The "Master-Slaves" Principle. Assigning control (system) functions to one master processor (Fig. 3,c) enables elimination of problems and shortcomings of the conventional MPS (protection of system data, reenterability of OS modules, complex connections in the equipment, inefficient use of processor circuits). Application and system tasks essentially use different instruction systems (especially in problem-oriented MPS's). Processor performance and efficiency can thus be enhanced by specialization of their internal structure. Moreover, since most interruptions go to the master processor, the "lag" of the slave processor can be raised; for instance, by increasing the degree of local parallelism or the size of the scratch-pad memory.

A drawback of rigid specialization of processors is the increase in their down times, since they cannot "help" each other as in the conventional MPS.

Pipeline, matrix, associative-matrix and flow processors can be used as the slave ones, as well as an entire subsystem of microprocessors [1,13,19,31].

Decomposition of System Functions. Realization of a uniform OS on a general purpose processor is very complicated (a set of programs with complex connections and a multilevel system of interrupts are used). If complete OS parts are functionally singled out and assigned to individual processors (Fig. 3,e), then the OS naturally breaks down into simple agents with standard connections, which greatly simplifies the design. As a result, the system can be made of simple microprocessors with a local memory, as provided in the UMASS design [34], or of special-purpose processors with a high-level internal language, as in the SYMBOL system [35]. In this system, besides the central processor there are seven off-line processors, with the following functions rigidly assigned to them: supervisor, main memory control, redistribution and purging of memory, disk control and listing, i/o channel control, program translation, editing and conversion of i/o information. These processors serve various application tasks at the same time, such as realizing type 1 parallelism.

A shortcoming of all asymmetric systems is the unbalanced nature of the load; i.e., given fluctuations in the characteristics of the job stream situations arise where all the tasks wait in a queue for one processor, while others are idle.

In a multiprocessor system [23], a less rigid specialization of processors based on dynamic microprogramming is used. Unevenness of the load of special-purpose processors can thus be reduced.

Collectivization of Hardware Resources. One way of raising operation execution speed is to use special-purpose functional (arithmetic) devices in the processor. This is most effective in the slave processor (Fig. 3,c). Using local parallelism enables time-matched operation of functional devices, but it is practically impossible to attain a high load of them [3,4,14,16,21]. This is related to the fact that the complexity of program analysis to reveal the local parallelism grows rapidly with an increase in the depth of analysis. The problem of loading functional devices in a multiprocessor system can be effectively resolved based on the principle of collectivization of devices [15,16,34,36], consisting of distributing the functional devices among command processors dynamically. In other words, they are collective resources (Fig. 3,h). The devices' operation is matched by paralleling different instruction flows (types 1 and 2 parallelism, and possibly 3).

A shortcoming of such a structure is that complex functions of device dispatch control and switching must be performed quickly. The principle is currently realized in the design of the HEP system [37].

The development of this approach is very promising. The system should include unconventional special-purpose devices, executing larger operators (for instance, computation of expressions and elementary mathematical functions, processing arrays and lists [28], etc.). Not only is the speed thus raised, but the machine's intelligence and level of its internal language go up, while the intensity of device interaction falls.

Systems With a Large Number of Small Processors (Two Examples). The design of the experimental system CHOPP seeks to create systems with up to a thousand processors, using parallelism of algorithm branches [20,38]. To provide dynamic redistribution of processors among short branches, the dispatcher functions are decentralized and embodied in the circuitry and microprograms. Each processor has a local supervisor to control intraprocessor processing and for interprocessor interactions. Several processors can redistribute task branches and memory pages at the same time, since the system tables and queues are decentralized and distributed.

A complex, fast commutator with an unconventional structure connects the processors with the system memory.

The low cost of microprocessors makes the construction of multimicroprocessor systems tempting [39]. However, technical constraints of modern microprocessors (low number of outputs, limited addressing possibilities, specific instruction system, lack of switching LSI circuit modules) hamper the building of general purpose multimicroprocessor systems.

A general purpose external computer is connected to the microprocessor system to perform most of the OS function. The structure of the central portion of the system is built on hierarchical, modular and similar principles. The system memory is hierarchical, with a nonhomogeneous access of the processors. In the

microprocessor system, the load on the resource control functions should not be lowered (for instance, by static adjustment of the system to the task), even at the cost of increasing microprocessor down times.

Let us look briefly at the experimental system Cm* [13,40,41], combining several hundred microprocessors (in 1980, the configuration included 50 microprocessors [42]). The system has efficiently solved various tasks (solving systems of equations, recognizing speech, translation) [13].

The system memory is represented to the programmer as a single field, but physically distributed. A local memory of 4 to 128,000 words is connected to each microprocessor. A commutation controller (made of MSI circuits) joins a group of up to 14 microprocessors into a module ("cell"), each controlled by a local operating system. The cells are connected with each other by buses (trunks) and commutation controllers, which buffer references to a "foreign" memory and convert addresses according to tables.

Concepts of Systems With Maximum Use of Intratask Parallelism. Using conventional sequential languages limits the possibilities of realizing internal parallelism of tasks. Languages of parallel asynchronous programming have been under development recently [43]; in particular, flow types (data flow), in which the operators are unordered, while the interactions among operators in terms of data and control are clearly specified. Such a language thus has no limits on the degree of use of parallelism. One possible MPS structure with an internal flow language is described in [31]. The system contains several central (processing) processors, several data structure controllers, a scheduling processor, peripheral processor and several types of internal memory.

Each central processor realizes parallelism of the operators of one procedure; i.e., the task fragment located in the local memory of this processor. The scheduling processor dispatches the procedures and central processors. The designs of the structures of the LAU and DDM1 flow processors are described in [29,30].

The concepts of modular asynchronous developed system (MARS) [5] and recursive computer system [44] are close to the flow systems. The basic principles of hardware structure, programming and control of the MARS system are asynchronism, hierarchical structure and modularity.

BIBLIOGRAPHY

1. "Mul'tiprotsessornyye sistemy i parallel'nyye vychisleniya" [Multiprocessor Systems and Parallel Computations], ed. by F.G. Enslow, M, Mir, 1976, 284 pages.

2. Golovkin B.A., "Parallel'nyye vychislitel'nyye sistem" [Parallel Computer Systems], M, Nauka, 1981, 520 pages.

3. Kartsev M.A., Brik V.A., "Vychislitel'nyye sistemy i sinkhronnaya arifmetika" "Computer Systems and Synchronous Arithmetic], M, Radio i svyaz', 1981, 360 pages.

4. Kartsev M.A., "Organization Principles of Parallel Computations, Structures of Computer Systems and Their Realization", KIBERNETIKA, 1981, No 2, pp 68-74.

5. Marchuk G.I., Kotov V.Ye., "Modul'naya asinkhronnaya razvivayemaya sistema (kontseptsiya)" [Modular Asynchronous Developable System (Concept)], Novosibirsk, 1978, 48 pages, 51 pages (SO AN SSSR, preprints No 86 and 87).

6. Enslow P.N., "Multiprocessor Organization: A Survey", Comput. Surveys, 1977, Vol 9, No 1, pp 103-129.

7. Kuck D.J., "A Survey of Parallel Machine Organization and Programming", Ibid., pp 29-59.

8. Baer J.P, "Multiprocessing Systems", IEEE Trans. Comput., 1976, Vol. 25, No 12, pp 1271-1277.

9. Critchlow A.J., "Generalized Multiprocessing and Multiprogramming Systems", AFIPS Conf. Proc. 1963, FJCC, 1963, Vol 24, pp 107-126.

10. Conway M.F., "A Multiprocessor System Design", Ibid., pp 139-145.

11. Leman M., "Parallel Processing and Parallel Processing Devices. Review of Problems and Preliminary Results", TIIER, 1966, Vol 54, No 12, pp 297-311.

12. Yevreynov E.V., Kosarev, Yu.G., "High-Performance Cellular General Purpose Computer Systems", Novosibirsk, SO Nauka, 1966, 308 pages.

13. "Multimicroprocessor Systems. Review and Example of Practical Realization", S.Kh. Fuller, Dzh. K. Usterkhut, L. Raskin et al, TIIER, 1978, Vol 66, No 2, pp 135-150.

14. Flinn M., "Superfast Computer Systems", TIIER, 1966, Vol 54, No 12, pp 311-320.

15. Aschenbrenner R.A., Flynn M.J., Robinson G.A., "Intrinsic Multiprocessing", AFIPS Conf. Proc., 1967 SJCC, 1967, Vol 30, pp 81-86.

16. Flynn M.J., Podvin A., "Shared Resource Multiprocessing", Computer, 1972, Vol 5, No 2, pp 20-28.

17. Flynn M.J., "Some Computer Organizations And Their Effectiveness", IEEE Trans. Comput., 1972, Vol 21, No 9, pp 948-960.

18. Golovkin B.A., "A Combined Classification of Computer Systems Based on Features of Parallelism of Information Processing", USiM, 1979, No 3, pp 19-28.

19. Thurber K.J., "Parallel Processor Architecture", Computer Design, 1979, Vol 18, No 1, pp 89-97; No 2, pp 103-114.

20. Sullivan H., Baskow T.R., Klappholz K., "A Large Scale, Homogeneous, Fully Distributed Parallel Machine", The 4th Annual Symp. Comput. Architecture Conf. Proc., 1977, pp 105-124.

21. Keller R.M., "Look-Ahead Processors", Comput. Surveys, 1975, Vol 7, No 4, pp 177-195.

22. Davis R.L., Zucker S., Campbell C.M., "A Building Block Approach to Multiprocessing", AFIPS Conf. Proc., 1972, SJCC, 1972, Vol 40, pp 685-707.

23. Murakami K., Nishikawa S., Sato M., "Poly-processor System Analysis and Design", The 4th Annual Symp. Comput. Architecture Conf. Proc., 1977, pp 49-56.

24. Rabinovich Z.L., "Development of Computer Structures in Connection With Enhancement of Machine Intelligence", USiM, 1976, No 6, pp 12-18.

25. Glushkov V.M., Pogrebinskiy S.B., Rabinovich Z.L., "Development of Structures of Multiprocessor Computers With Interpretation of High-Level Languages", USiM, 1978, No 6, pp 61-66.

26. Burtsev, V.S., "The Future of High-Performance Computer Creation", VESTNIK AN SSSR, 1975, No 3, pp 25-31.

27. Burtsev V.S., "Printsipy postroyeniya mnogoprotsessornykh vychislitel'nykh kompleksov "El'brus"" [Construction Principles of the "El'brus" Multiprocessor Computer Complexes], M, 1977, 53 pages (ITM and VT AN SSSR, preprint No 1).

28. "Mnogoprotsessornyye vychislitel'nyye sistemy" [Multiprocessor Computer Systems], M, Nauka, 1975, 144 pages.

29. Syre J.C., Comte D., Hifdi N., "Pipelining, Parallelism and Asynchronism in the LAU System", Proc. Intern. Conf. Parallel Processing. New York, 1977, pp 87-92.

30. Davis A.L., "The Architecture and System Methodology of DDM1: A Recursively Structured Data Driven Machine", Proc. 5th Symp. Comput. Architecture, 1978, pp 210-215.

31. Rumbaugh J., "A Data Flow Multiprocessor", IEEE Trans. Comput., 1977, Vol 26, No 2, pp 138-145.

32. Guntanis R.D., Viss N.L., "A Method of Selecting Processors to Handle Interruptions in Multiprocessor Systems", TIIER, 1966, Vol 54, No 12, pp 207-214.

33. Gonzalez M.J., Pamamoorthy C.V., "Parallel Task Execution in Decentralized Systems", IEEE Trans. Comput. 1972, Vol 21, No 12, pp 1310-1322.

34. Foster K.K., "The Direction of Computer Architecture Development", ZARUBEZHNAYA RADIOELEKTRONIKA, 1973, No 8, pp 42-52.

35. "SYMBOL: A Large Experimental System to Study the Possibility of Loading Software into Equipment", U.R. Smit, R. Ray, G.D. Chesley et al, KIBERNETICHESKIY SBORNIK. NOVAYA SERIYA, 1975, Vol 12, pp 115-149.

36. Karpman L.Ya., Yakuba A.A., "Investigation of a Structure of a Multiprocessor Central Computer", USiM, 1973, No 2, pp 89-94.

37. Smith B.J., "A Pipeline, Shared Resource MIMD Computer", Proc. Intern. Conf. Parall. Processing. New York, 1978, pp 6-8.

38. "The Node Kernel: Resource Management in a Self-Organizing Parallel Processor", H. Sullivan, T.R. Bashkow, D.K. Klappholz, L. Cohn, Proc. Intern. Conf. New York, 1977, pp 205-212.

39. Prangishvili I.V., Stetsyura G.G., "Mikroprotsessornyye sistemy" [Microprocessor Systems], M, Nauka, 1980, 238 pages.

40. "The Implementation of the CM* Multiprocessor", R.J. Swan, A. Bechtolsheim, K.W. Lai, J.K. Ousterhout, AFIPS Conf. Proc. 1977 NCC, 1977, Vol 46, pp 645-655.

41. "Problems of Programming Multiprocessor Systems", A. Dzhons, R. Chensler, Jr., A. Derem, TIIER, 1978, Vol 66, No 2, pp 151-163.

42. Jones A.K., Schwarz P., "The Experience Using Multiprocessor Systems: A Status Report", Comput. Surveys, 1980, Vol 12, No 2, pp 121-165.

43. Kotov V. Ye., "Parallel Languages", KIBERNETIKA, 1980, No 4, pp 1-10.

44. "Rekursivnyye mashiny i vychislitel'naya tekhnika" [Recursive Machines and Computer Engineering], V.M. Glushkov, M.B. Ignat'yev, V.A. Myasnikov, V.A. Torgashev, Kiev, 1974, 36 pages (IK AN USSR, preprint No 74-57).

45. Myasnikov V.A., Ignat'yev M.B., Torgashev V.A., "Rekursivnyye vychislitel'nyye mashiny" [Recursive Computers], M, 1977, 36 pages (ITM and VT AN SSSR, preprint No 12).

9875
CSO: 1863/196

UDC 681.325.5

DESIGN STRATEGY OF MICROCOMPUTER AND MICROPROCESSOR SYSTEM MEMORY

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 (signed to press 3 Jan 83, after revision 24 Feb 83) pp 11-15

[Article by Yuriy Sergeyevich Yakovlev, "The Strategy of Designing the Memory of Microcomputer and Microprocessor Systems", in the section "Microprocessor Technology and Its Application"]

[Text] The development of integrated technology has led to the creation of LSI circuits with high technical characteristics. With a degree of integration of up to 10^6 active components on a chip, where creation of a 32-bit computer is possible on one chip, microprocessor systems are becoming ever more predominant. Their main features are [1-3]:

extensive parallelism of the realization of the algorithm at all levels;

spatial distribution of the data processing means;

the ability to adjust the MPS (microprocessor system) architecture and structure to a specific algorithm and data structures, and others.

From these stem the special features of memory organization in the MPS:

from primarily concentrated, the memory becomes spatially distributed in MPS processors;

due to the flexible adjustment of the MPS, the need appears to create means for switching the "memory-memory" and "memory-processor" connections; i.e., the memory becomes alterable;

the functional capabilities of the memory expand; a tendency appears to add microprocessors to memory for solving local tasks within it (formation of the address code by a certain law; memory management; and so forth);

the basic MPS characteristics increasingly are determined by the memory characteristics; according to current estimates, the memory in a microcomputer amounts to up to 80% of all the hardware (without peripherals).

Along with the new quality features of memory, the need has thus appeared to solve the problem of organizing memory in a new formulation. The basis for solving this

problem is a new approach proposed by the author for designing memory, called the information-structural approach.

The essence of the method lies in the directed counter conversion (or selection) of the algorithm and data structures, on the one hand, and the structures of the hardware realizing the given algorithm, on the other, to establish a mutual relation between the algorithm, data structure and hardware structure enabling optimization of the given efficiency function, given the fulfillment of existing constraints on other parameters.

Consideration of the extensive parallelism of algorithm realization in the MPS, and the constantly growing degree of LSI circuit integration, have defined the need to single out four levels of parallelism: algorithmic level; task level; level of operators of high-level languages and machine instructions; and micro-operation level. Each parallelism level corresponds to the data units to be processed at this level, structural units of the hardware needed to realize the processing, and the corresponding rank of requirements on the technical parameters (TP) of these resources. The requirements are conditionally divided into system (TP_1), nodal (TP_2), local (TP_3) and partial (TP_4).

The system requirements basically define the direction of the ideology for building the memory architecture and structure as a whole, and the basic principles of its organization.

Requirements of the second rank (nodal) can be interpreted as those imposed specifically on a memory located in an MPS node and assigned to a specific processor. They must allow for the basic concepts of organization of the architecture and structure of microprocessors and microcomputers [4] and define the type ofmemory system (one- or multilevel), parameters (memory size, equivalent access time, etc.), features of the design (built into the microprocessor chip, located on the same board as it, etc.), and other characteristics needed to organize and use a specific memory system (for instance, protection of a certain array from unsanctioned access) within the MPS.

The third rank requirements (local) are those on memory modules forming the memory system, and on corresponding memory control controllers, as well as on external stores where needed (such as floppy magnetic disk stores).

Fourth rank requirements (partial) are those on specific LSI circuits and individual units (elements), on which the memory module or memory control module (controller) are built.

It should be emphasized that although the complexity of a unit of data representation, actions performed on the algorithm (program) and structural components of the system rises as one goes from a less significant level to a higher one, they are all uniform in terms of both form and rules for raising level complexity.

The considerations mentioned, as well as the existing hardware redundancy of LSI circuits and the structure of "memory-memory" and "processor-memory" connections, allow a generalized model of microcomputers and MPS's to be built (Fig. 1). It can serve as the basis for obtaining expressions establishing the relations between

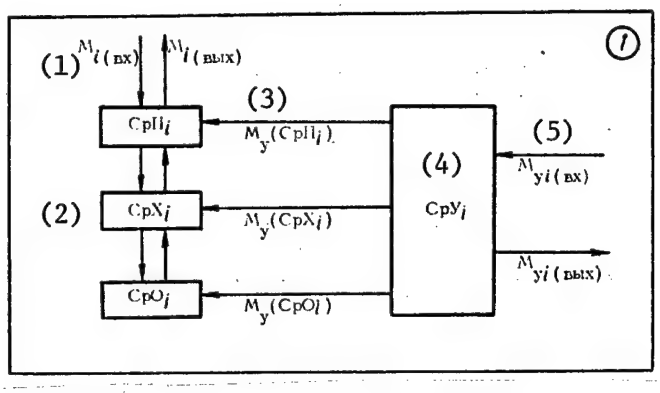


Fig. 1. Structural diagram of a model of microcomputers and MPS's for resource distribution: $M_{i(vkh)}$, $M_{i(vykh)}$, $M_{ui(vkh)}$, $M_{ui(vykh)}$ are the input [vkh] and output [vykh] buses of the information and control signals; $M_u(SrP_i)$, $M_u(SrKh_i)$, $M_u(SrO_i)$ are the buses of the control signals of the means for transmission, storage and processing, respectively.

Key:
1. $M_{i(vkh)}$, $M_{i(vykh)}$
2. SrP_i, $SrKh_i$, SrO_i
3. $M_u(SrP_i)$, $M_u(SrKh_i)$, $M_u(SrO_i)$
4. SrU_i
5. $M_{ui(vkh)}$, $M_{ui(vykh)}$

algorithm parameters, job requirements and requirements distributed among all varieties of hardware: processing (SrO), storage (SrKh) and transmission (SrP). A representation of the algorithm must be chosen and generalized, such as in the multilevel structure (YaPF) form developed by D.A. Pospelov [5].

The internal content of the nodes and arcs of the YaPF graph is defined by the level of algorithm representation considered: at the microoperation level, functional operators are the graph nodes; at the task level, the bodies of the tasks; at the upper level, the bodies of the task flows. The graph arcs are identified with the corresponding sets of input and output data. Each of the processes of the i-th level is embedded into the process of the higher (i-1) level, so that each YaPF node of the (i-1)-th level can be represented by the YaPF of the i-th level.

Considering the embedding of the processes, we represent the total realization time of the algorithm using the MPS for the synchronous method of executing sequential YaPF levels as:

$$Ta(c) = \sum_{l=1}^{4} \sum_{\mu=0}^{\delta_l - 1} \sum_{j=1}^{\varepsilon_{\mu l}} \tau_{\mu l j}^{(\psi_l)}, \qquad (1)$$

where δ_i is the number of levels of the YaPF of the i-th level; $\varepsilon_{\mu i}$ is the number of nodes in the longest (from the standpoint of realization time) sequence of the string of nodes of the μ-th level of the i-th YaPF level; $\tau_{\mu ij}^{(\psi i)}$ is the total realization time of the j-th node of the μ-th level of the i-th level of the YaPF by the μ_i-th set of hardware.

With the asynchronous principle of organizing processes, where there is no need to await the end of all the processes of the μ-th level to begin any process of the $(\mu+1)$-th level, in expression (1) appears a corrective coefficient $k_{\mu i}$, allowing for the reduction in T_a through the allowable additional parallelism of process realization.

An analysis of expression (1) shows that a reduction in T_a can be attained by: raising the internal computer language to the level of high-level language (YaVU) operators (the total number of levels is reduced); corresponding conversion of the YaPF for greatest paralleling, allowing for the functional capabilities of the hardware (the total number of levels μ declines); selection of the optimal hardware, whose set of operations corresponds to the required set of operations to realize the corresponding node of the YaPF graph (the parameter of $\tau_{\mu ij}^{\psi_i}$ is reduced); and optimal selection and distribution of MPS memory overall.

In this sense, the greatest performance effect can be provided if the memory system structure can flexibly redistribute its resources, including those of memory size, for the class of tasks realized. We shall call this property of the memory architecture and structure dynamic adjustment, and denote it as D_1. We shall then denote D_2 as static, not varying in the operating process, properties of the problem orientation of the memory architecture and structure (use of several ports, multisection structures, etc.). The appearance of microprocessors, a cheap and efficient means of data processing, has provided the possibility of dynamic conversion of data structures for their optimal adaptation to memory structure. We shall denote this property as D_3. Finally, one more memory feature must be singled out: expansion in the functional capabilities, denoted as D_4.

The microcomputer and MPS memory thus has a set of features $D \subset \{D_h\}$ and their related procedures, which we shall conditionally designate as $m(D) \subset \{m(D_h)\}$, $h=\overrightarrow{1,4}$.

Considering the parallelism of realizing algorithms, the set of U factors affecting the architecture and structure of microcomputer and MPS memory can be singled out:

$$U = \langle A_i,\ R_i,\ TP_i,\ P_{\ni i},\ m(D_h) \rangle\ , \qquad (2)$$

where A_i is the set of methods of representation, conversion and allocation of parallel sections (programs) at all levels of their representation; R_i is the set of algorithms of resource distribution at the i-th level of parallelism; TP_i is the set of parameters (requirements) of the hardware for storing information of the i-th rank of complexity; $P_{\ni i}$ is the constraints on the memory architecture and structure introduced by the component base and design possibilities of integrated technology.

The above makes it possible to propose the information-structural model of memory and the general strategy of memory design corresponding to it and to the parallelism levels:

$$\forall \Gamma_i = < A_i, G_i, TP_i, D_i > , \qquad (3)$$

where G_i is the set of architectures and structures of memory of the i-th rank; and D_i is the set of mappings of data and memory structures.

Based on (3), the following strategy of designing microcomputer and MPS memory is proposed:

$$\begin{array}{l}
\underset{\substack{a=\overline{1,3}\\ c=\overline{1,3}\\ h=\overline{1,4}}}{\forall}(\exists W: A_0 \quad G_0 \quad T\Pi_0, \Pi_0 \\
\quad\quad \downarrow \quad\quad \downarrow \quad\quad \downarrow \\
\quad\quad \downarrow \quad\quad \Pi_1 \quad\quad \downarrow \\
m_1(A_a) \rightarrow m_1(G_c) \rightarrow m_1(P_1 \rightarrow T\Pi_1) \rightarrow m_1(D_h) \\
\quad\quad \downarrow \quad\quad \Pi_2 \quad\quad \downarrow \\
m_2(A_a) \rightarrow m_2(G_c) \rightarrow m_2(P_2 \rightarrow T\Pi_2) \rightarrow m_2(D_h) \\
\quad\quad \downarrow \quad\quad \Pi_3 \quad\quad \downarrow \\
m_3(A_a) \rightarrow m_3(G_c) \rightarrow m_3(P_3 \rightarrow T\Pi_3) \rightarrow m_3(D_h) \\
\quad\quad \downarrow \quad\quad \Pi_4 \quad\quad \downarrow \\
m_4(A_a) \rightarrow m_4(G_c) \rightarrow m_4(P_4 \rightarrow T\Pi_4) \rightarrow m_4(D_h)), \\
\psi = \psi extr,
\end{array} \qquad (4)$$

where W is the mapping character of the algorithmic (procedural) description of memory onto the structural one; ψ is the criterion (set of criteria) of optimality; A_0 is the initial representation of the algorithm; G_0, the initial set of existing architectures and structures of the microprocessor equipment; TP_0, the initial system requirements for the microcomputer or microprocessor system as a whole; P_0, the basic initial construction principles, organization features and design constraints involving the microcomputer or MPS being designed. The strategy contains four identical levels (by type and sequence of procedures), completely agreeing with the levels of parallelism, requirements on the memory resources and level of their complexity.

The proposed model and multilevel strategy of memory design are the nucleus of the information-structural approach, including a system of interrelated algorithms for solving problems such as:

representation of an algorithm and its conversion to ensure the maximum possible parallelism at all levels of $m_i(A_i)$;

formation of initial sets of memory hardware for each level of $m_i(G_c)$;

development of the model, basic principles and technique of resource distribution at all levels of algorithm representation to determine the requirements of the corresponding ranks on the memory resources $m_i(R_i \rightarrow TP_i)$;

development of methods of representation and conversion of data structures and memory structures for their optimal mutual correspondence $m_i(D_h)$;

selection (development) of memory elements and organization on their basis of stores, modules and memory devices given high requirements on individual parameters (reliability, resistance to radiation and magnetic fields, etc.).

Typical procedures of conversion of information (an algorithm) are: $m_i(A_i)$, representation of the algorithm (program) in one of the convenient forms, such as the YaPF; $m_i(A_2)$, analysis and evaluation of the algorithm to reveal standard data structures, determination of conversions of data structures at all levels of parallelism, enabling their optimal location in the memory and efficient utilization during algorithm realization; $m_i(A_3)$, allocation of subsets of data structure conversion procedures, which should be assigned to one or more microprocessors specially allocated for this, and development of the algorithm for realizing the allocated procedures.

Standard procedures in forming sets of hardware for resource distribution, including those of memory, are: $m_i(G_i)$, formation of the initial library of standard conventional architectures and structures of hardware for each rank of requirements; $m_i(G_2)$, formation of a library of basic versions of hardware architectures and structures, allowing for the particular features of integral technology and its development trends; $m_i(G_3)$, definition of the basic parameters of the hardware and formation of special criteria for their directed analysis and selection.

The essence of procedures defining requirements on the resources for storing information $m_i(R_i \rightarrow TP_i)$ can be formulated as follows: at each design strategy level, find the optimal relation between the parameters of the algorithm realized and the required memory parameters, together with the parameters of the remaining hardware, such that the values of one or more parameters of the microcomputer or microprocessor system are minimized (or maximized) according to the quality criterion, given fulfillment of constraints on other parameters given in the specifications. The solution of this problem should be done in the following three stages to reduce the labor input.

The first stage can be called the preliminary one. At it, the most acceptable ranges of required values of the storage hardware parameters are defined, together with other hardware in the microcomputer and microprocessor systems. The basis of the strategy here consists of: a developed, generalized information-structural model of the microcomputer and microprocessor system for resource distribution, presented in Fig. 1; formula dependences, obtained for this model and reflecting the relation of the basic parameters of the algorithm and the corresponding

parameters of the hardware required to realize this algorithm [6]. As a result of performing this stage based on the initial hardware sets N_{ij}, the basic sets $N_{(b)ij} \in N_{ij}$ are formed for each type of hardware and each level of design strategy [6].

The second stage is the basic one. At it, for each level of the design strategy there is established a unique relation between the algorithm parameters and those of the hardware in the basic sets $N_{(b)ij}$. The basis of this stage's strategy is: the criterion of evaluating the functional capabilities of each of the technical resources of the sets $N_{(b)ij}$ and their relation to the algorithm; a directed search algorithm of such a relation; the time load matrix of each of the technical resources used of sets $N_{(b)ij}$, intended to determine the realization time of the corresponding section (according to the representation level) of the algorithm; a time diagram of the operation of each of the resources to reveal conflict situations arising from simultaneous request for common resources by several resources. As a result of fulfilling the second stage, to each hardware resource of the sets $N_{(b)ij}$ there should be assigned, at each algorithm representation level, certain sections (fragments, YaPF graph nodes, etc.) of the algorithm, which are realized most efficiently using this resource, and the time load revealed of each of the resources with allowance for possible conflict situations. At this stage is defined the total algorithm realization time according to (1), and fulfillment checked of all the constraints given in the design specifications.

The third stage of resource distribution is the refining one, and is intended for fuller optimization of the system being designed by revealing parameter value reserves on which there are constraints in the specifications, and resource redistribution according to it. The basic strategy here is the method of directed inspection of existing system resources, as well as the relations obtained for evaluating the change in parameters of the hardware and the corresponding changes in the values of the optimized parameter. The result of fulfilling the third stage of resource distribution at each level of the memory design strategy is the architecture and structure of each level's hardware, which is an integral part of that of the higher level, along with the set of technical requirements on the storage resources of the level examined and of higher ones [6].

An analysis of existing hierarchical structures of memory and features of their organization at the current component level enables a generalized, standard model of a hierarchical memory system to be suggested. With the most common number of levels (three), it coincides with the generalized information-structural model of microcomputers and MPS's for resource distribution (Figs. 1, 2). This, in turn, has made it possible to create an engineering technique for designing such systems, according to which the values of the basic parameters of the hierarchical memory system can be defined by a modelling system that does not require a great deal of time, using rather simple formula dependences [7]. The good coincidence of experimental and calculation data confirms the correctness of the initial concepts of constructing the hierarchical memory system model.

Solving the questions of the next level of general memory design strategy (4) is related to development of theoretical foundations, structure organization and the design of various memory modules.

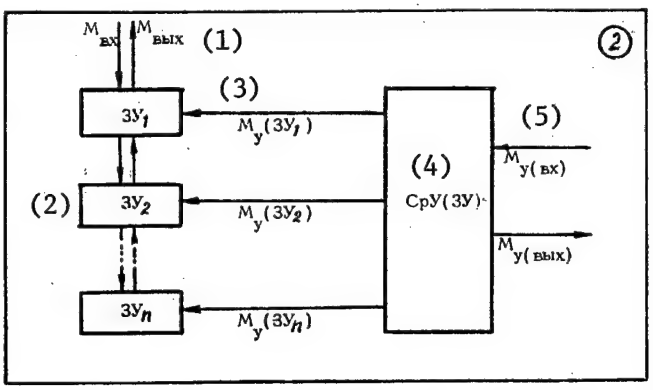


Fig. 2. Structural diagram of an equivalent model of a hierarchical memory system: SrU(ZU), the device controlling the memory system; $Mu(ZU_1),\ldots, Mu(ZU_n)$, the buses of the control signals $ZU_1,\ldots, ZU_n$, respectively.

Key:

1. M_{vkh}, M_{vykh}
2. ZU_1, ZU_2, ZU_n (memory unit 1, 2,...,n)
3. $M_u(ZU_1)$, $M_u(ZU_2)$, $M_y(ZU_n)$
4. SRU(ZU)
5. $M_{y(vkh)}$, $M_{y(vykh)}$

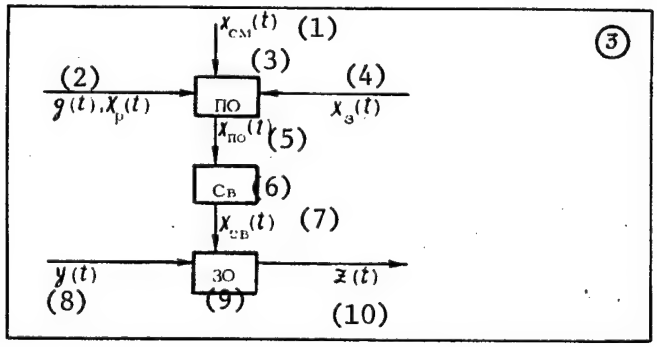


Fig. 3. Simplified model of a total current memory element made on multiaperture ferrite plates (MFP) and thin magnetic films (TMP): $x_z(t)$, $x_r(t)$, $g(t)$ and $x_{sm}(t)$ are the address, bit and setting signals of recording and the bias signal, respectively; y(t) and z(t), the address read signal and output intelligence signal; PO, ZO, Sv, the switching and memory working and region of communication among them of the memory element on MFP; $x_{po}(t)$ and $x_{sv}(5)$, signals appearing during switching of the PO regions and the region of connection between PO and ZO, respectively.

Key:

1. $x_{sm}(t)$
2. $g(t), x_r(t)$

3. PO
4. $X_z(t)$
5. $X_{PO}(t)$
6. Sv
7. $x_{sv}(t)$
8. y(t)
9. ZO
10. z(t)

Special attention should be focused on solving the problem of enhancing the efficiency of using semiconductor memory devices made on IGFE-LSI dynamic type circuits by reducing loss of system performance from the need to perform data regeneration.

Preservation of information when the power sources are turned on, low power consumption in the storage mode, high resistance to various electromagnetic and radiation fields--all these characteristics, together with the integrated technology of manufacturing, are decisive for the use in microcomputers and MPS's of memory modules made on multiaperture ferrite plates (MFP) and thin magnetic films (TMP). Research performed in creating new types of memory elements and memory modules on MFP and TMP has enabled such basic principles of their organization to be formulated as the third beat principle, fourth bus principle, separate bus control principle, word current inhibition principle, principle of adding the current given a bias, and others.

These principles have enabled construction of a simplified model of a total current memory element made on MFP and TMP (Fig. 3), whose basic features coincide with the generalized information-structural model of microcomputers and MPS's in Fig. 1. This confirms the generality of the information-structural approach to memory design at all levels of the proposedstrategy. Further conversion of the memory element model and its reduction to a simple automat model have defined the methods of synthesis of total current memory MFP and TMP elements, and form the basis for making memory elements, as well as a new type of storage devices having higher parameters of functional reliability and speed [8].

The proposed information-structural approach thus enables a relation to be established between the algorithm realized, data (information) structures, and structure of the memory system as an integral, major part of a problem-oriented microcomputer and microprocessor system.

Based on a foundation of three theories, of algorithms, automata and creation of data storage tools, the information-structural approach provides solution to the important scientific problem of developing the theoretical foundations, organization principles and design methods of new types of systems and modules of memory with an adaptable architecture and structure and enhanced functional capabilities for problem-oriented microcomputers and microprocessor systems.

BIBLIOGRAPHY

1. "Rekursivnyye mashiny i vychislitel'naya tekhnika" [Recursive Machines and Computer Technology], V.M. Glushkov, M.B. Ignat'yev, V.A. Myasnikov, V.A. Torgashov, Kiev, 1974, 26 pages (IK AN USSR, preprint No 74-57).

2. Glushkov V.M., Nikitin A.I., Rabinovich Z.L., "Some Development Trends in Digital Computer Structures and Software", USiM, 1972, No 1, pp 79-85.

3. Glushkov V.M., Pogrebinskiy S.B., Rabinovich Z.L., "Development of Multiprocessor Computer Structures with Interpretation of High-Level Languages", USiM, 1978, No 6, pp 61-66.

4. Palagin A.V., "Development Prospects and Questions of the Theory of Microprocessor and Microcomputer Design", USiM, 1982, No 3, pp 24-29.

5. Pospelov D.A., "Vvedeniye v teoriyu vychislitel'nykh sistem" [Introduction to Computer System Theory], M, Sov. radio, 1972, 280 pages.

6. Yakovlev Yu.S., "O raspredelenii resursov pamyati mikroEVM" [Distribution of Microcomputer Memory Resources], Kiev, 1982, 47 pages (IK AN USSR, preprint No 82-65).

7. Yakovlev Yu.S., "Construction of Hierarchical Systems of Microcomputer Memory", USiM, 1980, No 5, pp 16-21.

8. Malinovskiy B.N., Yakovlev Yu.S., Makovenko Ye.T., "Analiz i sintez mini-EVM" [Minicomputer Analysis and Synthesis], Kiev, Naukova dumka, 1976, 176 pages.

9875
CSO: 1863/196

UDC 681.3.008

ORGANIZATION OF PROGRAM CHANNEL OF DATA FILE INPUT-OUTPUT IN SYSTEMS BASED ON K580IK80 MICROPROCESSORS

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 (signed to press 25 Jan 82, after revision 26 May 82) pp 16-19

[Article by Mikhail Petrovich Naumchik, "Organization of the Program Channel of Data File Input-Output in Systems Based on K580IK80 Microprocessors", in the section "Microprocessor Technology and its Application"]

[Excerpts] In the development of microprocessor systems, the task frequently arises of organizing the data file input-output channel (KVVMI) to meet the requirement of optimal combination of minimal equipment cost and ensuring maximum possible channel speed.

Organizing the direct access to memory channel provides high speed, but still has drawbacks limiting the range of this channel's application (notably, the amount of equipment of the microprocessor system increases).

An alternative to this KVVMI organization can be a program channel providing data input-output using a special program (subroutine), requiring minimal equipment cost.

The instruction system of the K580IK80 microprocessor includes the special instructions IN and OUT, designed to enter and output data through an addressable port. Using these instructions when organizing the program KVVMI does not affect the addressable region of the microprocessor memory (64 Kbytes).

The second approach to making a program channel lies in using an external device to separate part of the addressable region of memory, enabling use of any instructions providing access to memory for input and output [1]. This helps to considerably increase the input-output rate, while at the same time reducing the memory required to store the program. The second approach will thus be used below in considering methods of organizing the program KVVMI. Since the organization of input and output processes is identical in many respects, we shall consider only data file output algorithms.

Using the above-described methods of building the program KVVMI enables the desired results to be achieved in several practical applications, with low memory cost of the microprocessor system and a negligible microprocessor load.

The input-output programs, using stack instructions, can be widely used in organizing fast transmission of data files within the microprocessor system; for example, from ROM to main memory, during data input-output from A/D and D/A converters, and during data exchange between systems.

BIBLIOGRAPHY

1. INTEL-8080. Microcomputer Systems Users Manual, INTEL Corp., 1975, 240 pages.

2. Bedrekovskiy M.A., Kruchinkin N.S., Podolyak V.A., "Mikroprotsessory" [Microprocessors], M, Radio i svyaz', 1981, 72 pages.

3. "Analysis of Data Display Devices for Interface with Minicomputers", A.I. Sigalov, A.V. Palagin, V.V. Badashin et al, USiM, 1977, No 1, pp 88-94.

9875
CSO: 1863/196

UDC 681.32

SELECTION OF BIT-SLICE MICROPROCESSOR STRUCTURE

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 (signed to press 9 Nov 81, after revision 30 Dec 82) pp 21-24

[Article by Viktor Vladimirovich Novoselov and Lev Alekseyevich Shumilov, "Selection of Bit-Slice Microprocessor Structure", in the section "Microprocessor Technology and Its Application"]

[Excerpts] Introduction. The basic principle in designing high-performance digital devices is to correlate the device's structure with the algorithm performed. In this work, this principle is specified to the level of quantitative estimates relative to the task of selecting the structure of a microprocessor made on a set of microprogrammable sectionalized microprocessor LSI circuits. The particular feature of designing bit-slice devices (in contrast to those based on single-chip microprocessors) is that,before programming, the designer must make basic decisions on the organization of the microprocessor's hardware part and the corresponding microsoftware [1-4]. Using the quantitative estimates of the degree of correlation between structure and algorithm proposed below enables a definition of the optimal microprocessor structure to the stage of microprogram compilation.

In conclusion, it should be noted that the special features of the architecture of certain LSI circuits (such as the K589IK01) narrow down the diversity in standard microprocessor structures realized. The modern set of microprogrammable microprocessor LSI circuits K1804 has the broadest capabilities from this standpoint.

BIBLIOGRAPHY

1. Mick J., Brick J., "Bit-Slice Microprocessor Design", New York, McGraw-Hill Book Co., 1980, 398 pages.

2. Myers G.J., "Digital System Design with LSI Bit-Slice Logic", New York: John Wiley & Sons Inc., 1980, 338 pages.

3. Alexandridis N.A., "Bit-Slice Microprocessor Architecture", Computer, 1978, No 6, pp 56-79.

4. Vanneschi M., "Microprogrammable Processor Components and Architectures", The Microprocessor and Its Application (An Advanced Course), Cambridge, University Press, 1978, pp 219-253.

5. Mayorov S.A., Novikov G.I., "Struktura elektronnykh vychislitel'nykh mashin" [The Structure of Computers], L, Mashinostroyeniye, 1979, 384 pages.

6. Schecher H., Anlauff H., "Avoiding Gaps in a Three-Level Pipelined Micro-Control Unit", Microprocessing and Microprogramming, 1982, Vol 9, No 3, pp 155-160.

9875
CSO: 1863/196

UDC 681.327.8

COMMUNICATION PROCESSOR BASED ON SM-1 MINICOMPUTER IN YeS COMPUTER DATA TELE-PROCESSING SYSTEM

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 (signed to press 22 Apr 82, after revision 21 Sep 82) pp 28-33

[Article by Yuriy Musovich Bishenov, "A Communication Processor Based on the SM-1 Minicomputer in a YeS Computer Data Teleprocessing System", in the section "Microprocessor Technology and Its Application"]

[Text] Introduction. A central link in the structure of data teleprocessing system (STD) hardware of YeS computers are terminal multiplexers (MPD), providing the interface between the computer and the data transmission equipment, and linking the computer with remote terminals (AP). The MPD can be conveniently divided into three groups [1,2]: hardware MPD's; programmable MPD's; and connected processors (data teleprocessing processors).

Hardware multiplexers are represented in the YeS computers by domestic models MPD-1A (YeS8400) and MPD-3 (YeS8403). They perform the following basic functions [1,2]:

realization of central computer programs and formation of status bytes;

realization of algorithms of interaction with the YeS computer multiplex channel through a standard input-output interface;

realization of various algorithms controlling the exchange of data defined by the terminals connected to the MPD;

realization of algorithms of interaction with the data transmission equipment and communication channels;

data transmission at varying speeds; and

monitoring the correctness of data received.

Programmable multiplexers perform the same functions as hardware ones, but in contrast to them can be quickly adapted to another operating algorithm by connecting new terminal types to them. Switching to a microprogram or program control principle in these MPD's by ordinary rewriting of a microprogram into permanent memory or

a program into main memory of the multiplexer allows data exchange algorithms to be modified without changing the hardware, and new exchange algorithms to be added when expanding the terminal equipment.

Examples of microprogram MPD's are models MPD-2 (YeS8402) and MPD (YeS8404). Another familiar example is the programmable MPD based on the YeS1010 minicomputer, simulating the functions of the hardware MPD [3].

Flexible adjustment to new operating conditions enables the programmable MPD to lower equipment costs, but it is slower than the hardware one. This is the basic reason for the appearance of the combined (hardware-microprogram) principle of control in the MPD.

A promising direction in STD development is the use of communication processors in them, which represent the next stage in MPD development. Their distinguishing feature is performance of partial processing of a message, which can include the following functions [1,4]:

realization of the algorithm of access of the central computer to the terminal, and freeing the computer from routine operations;

concentration of messages, their analysis, check, processing and transmission to the appropriate terminal;

message switching;

control of the operation of individual terminals;

accumulation, buffering and storage of data, and various data transmission network control functions.

These functions are usually joined together in a special message control program [4].

An example of this class of multiplexers is the YeS8371 data teleprocessing processor, developed in Bulgaria as the basic link in the YeSTYeL-4 STD [5]. There are no such comparable devices in the range of serially produced Soviet data teleprocessing devices.

The "Kolos" STD, with a radial structure, was developed to create an ASU [automated control system] of agricultural supply of the Kabardino-Balkarsk ASSR [6]. At the next stage, the "Kolos" STD will grow into a computer network with a radial-ring topology. This has led to the need for a specialized computer with its own software, which would perform the role of the connecting link between the YeS1022 information computer and the terminals.

This article describes a communication processor based on the SM-1 minicomputer in the YeS computer data teleprocessing system.

General Structure of the Communication Processor. The contents and configuration of this communication processor's hardware are given in a figure on the inside back cover.

An A-711 channel matcher joins the standard interfaces 2A (YeS computer) and 2K (SM-1 i/o interface). The matcher is connected to the SM-1's direct memory access selector channel (KPDP), and provides multicomputer operation of the YeS1022 and SM-1 at the "channel-channel" level.

Besides the A-711, the KPDP connects two high-speed NMD's [magnetic disk storages], the size of one of which compensates the lack of main memory of the SM-1 i/o buffers.

The A-721 matchers are used to connect the "APD-MA--telephone channel" data transmission equipment to the SM-1 on a radial scheme. The program channel, formed by the SM-1 processor and the A-721 matchers, in combination with the minicomputer interrupt system enables maximum combination of data i/o operations from low-speed communication channels.

The total service time of one enquiry of the A-721 device with the describable program organization of input-output is not more than 75 microseconds, corresponding to a throughput of $2 \cdot 10^5$ bit/s. At a 1200 bit/s data transfer rate via telephone channel, over 100 A-721 devices can be connected.

As the remote terminals are used ones that are a connected chain of APD-MA, "Videoton-340" and "Iskra-2302" devices [6-8], called the single-point version of the terminal. The addressable element in the terminal is the "Videoton-340" display. The terminal operator switches the contents of the display buffer to the "Iskra-2302" and back.

The multipoint terminal version consists of several single-point ones, and is supplemented by an addressing unit, which inserts the address byte during transmission from the terminal to the line and selects the necessary component by the first byte of the message received from the line.

Software Structure. The software of the communication processor on the SM-1 is used to support the information computer's message control program, which performs functions of the general telecommunications method of OS YeS access and was specially developed for the "Kolos" STD. In the YeS computer, the interface with the SM-1 is in the form of a package designed for the graphic method of OS YeS access.

The contents of the communication processor's software are given in an illustration.

The programs RCV.YeS and RCV.AP receive information from the YeS computers and the remote terminals, respectively. The programs SND.SM and SND.AP send data to the YeS computers and remote terminals, respectively.

The RW module provides operation with queues of input messages IQ and output messages OQ. These queues have the same organization, and are sets of direct access data on the SM-1 magnetic disk storage. The RW module realizes dynamic list allocation of external memory of the message queues. The need to use a disk memory stems from the insufficiency of the working memory of the i/o buffers (the SM-1 working memory volume used is 32K bytes).

The dispatcher of program interaction COMM is designed to synchronize reception-transmission programs, buffer messages, and serve IQ and OQ queues, fulfilling the role of matchers (of varying transmission speed) of the YeS1022 selector channel and telephone communication channels. The COMM module organizes two messages flows: from the YeS1022 host computer to the remote terminals, and vice versa.

All the modules described operate in the multitask DOS of the SM-1, supporting functions of the macroinstructions WAIT, POSTE, EXIO, STIME, RUN and TURN [9] and including the direct method of access by relative address of block on magnetic disks.

There is presently a tendency to view teleprocessing software in a comprehensive manner, allowing for the "hard" programs by which the STD hardware operates. Another, related tendency is to divide the software into functional hierarchical levels. At the upper levels are the application programs of processing; at the lower ones, the "hard" programs of the various hardware forming the physical data transmission channel.

Since the main function of each STD program level is data transmission, STD software should be classified by levels of data transmission channels [10]: physical channel; information channel; network of information channels; logic channel; and network of logic channels.

In the ascending hierarchy of channels, the communication processor's software usually includes the first three. The multilevel principle of design, widely used in generating computer network software, considerably eases the design and maintenance of software.

Below are described the basic components of a communication processor based on the SM-1, based on the above principles.

Information Channel of the Parallel Physical Channel. The information channel forms the second STD software level, and is interfaced with the physical channel level. Its protocol performs error-free reception (transmission) of a message via the information channel, and performs the following functions:

physical channel activation and deactivation;

definition of the director of the vector of transmission through the half-duplex physical channel;

reception of acknowledgment of reception of message portions;

retransmission of unreceived message portions, meaning those for which a negative acknowledgment of reception is obtained;

and generation of check symbols during transmission, enabling reception or transmission errors to be monitored.

The information channel depends on the type of physical channel (series or parallel), and also on the selection of the control method with the series physical channel: byte data transmission (byte-oriented protocol), transmission of bit flows not divided into bytes (bit-oriented protocol), or absence of control.

The protocol of the high-speed information channel is designed to control the "SM-1--YeS-1022" parallel physical channel. It differs from a similar protocol in [10] in that the specific features are considered of the i/o driver supporting the A-711 matcher. The protocol emphasizes the dependence of the SM-1 telecommunication processor on the information one in the YeS computer.

Interaction of two computers connected to the ends of the parallel physical channel requires an exchange of control instructions, followed by transmission of the information message in the required direction.

The protocol described uses the following instructions: SR, entry request (SM-1 or YeS1022 reception proposal); SW, output request (transmission proposal); S^+, positive acknowledgement of reception; S^-, negative acknowledgment of reception (message retransmission proposal); SS, end of communication (completion of exchange protocol); TEXT, information frame consisting of two parts, a two-byte prefix, and an information part. The two-byte prefix defines the length of the frame's information part.

Further description of the high-speed information channel protocol should be done in the three following phases:

establishment of connection (communication);

transmission of information frame; and

disconnection.

In the first phase, the communication between the two machines (SM-1 and YeS1022) is established, and the direction of the vector of transmission through the physical channel defined. In the protocol described, this phase begins when the instruction SR (invitation to transmit) or SW (request for transmission) is sent from the Yes1022 (YeS side). If the SM side can execute the requested instruction, it answers with a different packet (SR for SW, SW for SR). Otherwise, the SM side responds with the same packet (SR for SR, SW for SW).

The initiative for establishing the connection in this protocol thus belongs to the YeS side. A connection is considered unestablished if the same SR or SW instructions encounter each other.

We shall now look at four possible situations in the connection establishment phase, shown in a figure on the inside back cover, Fig. 3.

Fig. 3a shows a situation where the YeS side has invited the SM-1 to transmit (SR instruction), but the SM has no data to send (SR instruction). The connection is not established, and the transmission vector not defined.

In Fig. 3b, the YeS side has invited the SM-1 to transmit (SR instruction), and the SM side answers with a request for transmission (SW instruction). The connection is established; the transmission vector, YeS-SM. After receiving the SW instruction, the YeS side responds with the SR instruction. This is related to restoration of the symmetry upset by the fact that the initiative for connection always belongs on the YeS side.

In Fig. 3c, the YeS side has requested transmission and the SM side is ready to receive the message text. The connection is established; the transmission vector, SM-YeS.

In Fig. 3d, a situation is considered where the YeS side has requested transmission (SW instruction), but the SM side cannot satisfy the request (SW instruction). The connection is not established, and the transmission vector is not defined.

Of the situations described, only those in Fig. 3b and c end with the connection established. The situation in Fig. 3a occurs when neither side is active, and the situation does not require resolution. The situation in Fig. 3d is a conflict one, whose solution uses the following synchronizing rule: in the next phase of connection establishment, the SM side gives the SR instruction to support the YeS side's transmission request.

The second phase starts after the connection is established. The basic task of the phase is error-free transmission of the message text in the direction of the transmission vector established. The transmitting side (at the beginning of the transmission vector) transmits the message text. The receiving side, having received the text, checks it for errors. If it finds them, it gives the S^- instruction to request retransmission.

The check of the text received on the receiving side consists of the coincidence of received bytes with the message prefix and the absence of parity errors of the information bits of the byte.

Having received the S^- instruction, the transmitting side retransmits the message. This cycle continues until the message is received without errors, or the number of repetitions exceeds a certain defined time-out.

The third phase begins after error-free transmission of the message. It consists of the receiving side sending the instruction S^+ (positive acknowledgement), in response to which the transmitting side sends the packet SS (end of communication), which completes the data exchange protocol.

The protocol of the high-speed information channel is realized by the modules SND.YeS and RCV.SM (based on the OS YeS graphic method), and SND.SM and RCV.YeS (based on the A-711 support driver).

Information Channel of the Series Physical Channel "SM-1--terminal". According to the classification in [11], controlled and uncontrolled transmissions through the channel are distinguished. The basic characteristic of the controlled transmission is that the receiving side knows that the message is being sent. This is done by the receiving side sending a special polling signal, requiring the transmitting side to send either a message or a signal that there is not one.

In uncontrolled transmission, the receiving side cannot affect the time when the message is sent.

The choice of one or another transmission control method is determined by the specific features of the STD hardware, or to be more precise, that of the terminals. Controlled transmission is achieved by complicating the terminal equipment, since the terminal (or rather, the terminal control device) requires additional logic circuits reacting to the polling signals. In uncontrolled transmission, the equipment of the channel end opposite the terminal becomes more complex, since the priority of message reception from the terminal must be resolved.

Controlled transmissions are most effective on high-speed channels, but require additional expense due to the specific features of remote terminals. The polling signals can comprise a major portion of the transmission channel's total traffic.

Uncontrolled transmissions require reception priority at the channel end opposite the terminal, but there are usually sufficient resources for resolving this priority.

A combination of these transmission methods obviously provides greater effectiveness in solving various ASU tasks by teleprocessing methods.

The series physical channel "SM-1--terminal" described can be classified as uncontrolled, which leads to the need to resolve priority reception of messages from remote terminals in the SM-1.

The protocol for the series information channel is realized by the SND.AP and RCV.AP modules, based on the A-721 support drivers. The special feature of this protocol stems from the need for priority entry of messages from the terminals, achieved by calling the RCV.AP module from the SND.AP; i.e., activation of the SND.AP module can be completed by both transmission and reception of the information message.

Message Addressing. The design of the communication processor software used the following concept of message addressing, allowing for the nature of the tasks solved.

The message sources and receivers are different objects: information computer message control program, application program, terminal. Each object is addressed by a special common 16-bit address, structurally divided into a five-bit computer index (of the information computer or communication processor), serving the object, and an 11-bit displacement, directly choosing the object itself.

The communication processor serves a certain set of objects (such as terminals), connected to it physically, and converts their physical addresses to the displacement and vice versa. Some objects may not be directly connected to the communication processor, as when there is one information computer or during transition from a radial network topology to a radial-ring one. In this case, the communication processor defines the physical address of the line leading to this object by the common address index.

The communication processor must therefore have two tables: one of indexes, and one of objects. The input to the latter is achieved by the displacement of the common address, if its index coincides with that of the communication processor. The output of each table is a list of channels leading to the object where the index or displacement of the common address are used for access to the table. The COMM dispatcher chooses the first available channel from this list.

Thus, the message always has two common addresses: of the source object and the receiver object. Some objects, such as terminals, cannot generate a common address of the receiver object in the message in the required format. The communication processor then assigns these messages a common address of the message control program of the information computer, which serves the given communication processor. The message control program, interpreting certain fields of this message, defines the common address of the receiver object's application program, or sends the message to one of the processing routines.

The information message is usually packed into one or more information packets. Besides the two common address fields, each packet contains an eight-bit service field, whose first bit (the mesh bit, set at zero) defines the multipacket message. The remaining seven bits are used to number the packets, which limits the maximum message length to 128 packets. The first packet of the message has the number 0.

In the software version described, the packet length is still undefined; it is a variable amount. All the messages are assumed to be one-packet ones (zero value of the mesh bit and zero number of the package). This is why there are not packet assembly and disassembly modules. Moreover, flow control functions are not performed in the communication processor, since it is assumed that an operating mode will be used which carries these functions to the information computer [10].

The COMM Dispatcher's Operating Algorithm. To describe its operation, it helps to assign each of the four simplex information channels one buffer each. The COMM status is thus determined by those of these four buffers, each of which can have one of two states: busy or free.

The COMM status can be coded by four bits, one for each buffer. Fig. 4 shows the COMM's constitution diagram. It uses the following correlation between bits and buffers of the information channel: first bit, information channel between YeS and SM; second bit, between SM and terminal; third bit, between SM and YeS; and fourth bit, between terminal and SM.

This diagram shows a binary representation of the status, and its decimal code. Only the junctions between the states are shown, without indicating the input signals causing these transfers, and without indicating the output signals generated by them. The junction from one state to another (the presence of an arc) corresponds to a response by one of the four reception-transmission automats. The presence of binary units in the status means COMM activity: the automaton will try to send the information message and free the buffer (convert the unit to zero).

It can be seen in Fig. 4 that the status of the COMM dispatcher $0000_2=0_{10}$ occurs, for example, when all the buffers are free and the dispatcher can adjust for

reception via the information channels "YeS-SM" and "terminal-SM". The status $1111_2=15_{10}$ is the last of the possible 16 states of the COMM dispatcher. All the buffers are busy, and none of the information channels of reception can be activated. In this state, the COMM periodically starts the information channel of transmission "SM-terminal" or "SM-YeS", and tries to move to easier states in the binary configuration.

Another typical state is 1000_2, occurring after the "YeS-SM" information channel responds in the 0000_2 status. The dispatcher performs the exchange of this channel's buffer with that of the information channel "SM-terminal", and goes to the 0010_2 state, in which three information channels can operate: between YeS and SM, between SM and terminal, and between terminal and SM.

The COMM dispatcher thus performs binary buffering during transmission of messages in two directions: from terminal to YeS, and from YeS to terminal.

The constitution diagram is obtained by ordering the states by number of exiting arcs. The arcs correspond to filling or freeing of the reception-transmission buffer. For instance, four arcs exit the $0110_2=6_{10}$ state; i.e., all four simplex information channels can operate.

States with three outputs are the most typical of the COMM diagram. There are also two states with two outputs $0000_2=0_{10}$ and $1111_2=15_{10}$, examined above. States with one output arc are the simplest. The only action performed by the COMM dispatcher in this state is to replace the input buffers with output ones, and vice versa.

Generation of Communication Processor Software. The basic language for realizing the programs SND.YeS, RCV.YeS, SND.AP and RCV.AP and the COMM dispatcher is FORTRAN. There is a translator from FORTRAN on all SM computers, which makes the communication processor software more portable. At the same time, programming in FORTRAN has enabled certain programs in the OS YeS medium, having advanced debugging tools, to be debugged. The RW module is in the SM-1 assembler, since high requirements are imposed on it.

BIBLIOGRAPHY

1. "Mul'tipleksory peredachi dannykh" [Data Transmission Multiplexers], ed. by V.S. Lapin and A.I. Korchinskiy, M, Energiya, 1980, 160 pages.

2. Pozin I.L., Shcherbo V.K., "Teleobrabotka dannykh v avtomatizirovannykh sistemakh" [Data Teleprocessing in Automated Systems], M, Statistika, 1976, 180 pages.

3. Shugar P., "The Software System of a Programmable Multiplexer Based on the YeS1010 Computer", VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN, 1978, Vol 4, pp 105-111.

4. N'yuport S.B., Ryzlyak Ya., "Communication Processors", TIIER, 1972, Vol 60, No 11, pp 70-83.

5. Zlatov Z., Mikhaylov M., Matrozov A., "The YeS8371 Data Teleprocessing Processor", VYCHISLITEL'NAYA TEKHNIKA SOTSIALISTICHESKIKH STRAN, 1981, Vol 10, pp 171-173.

6. Ivanov P.M., "The "Kolos" Data Teleprocessing System on the Unified Series [YeS] Computer", USiM, 1982, No 2, pp 75-78.

7. "The Terminal "APD-MA--Videoton-340--Iskra-2302"",Ivanov P.M., Gridyakin I.I., Tkhishev N.A., et al, USiM, 1980, No 2, pp 137-139.

8. Ivanov P.M., "ASU proizvodstvenno-tekhnicheskim obespecheniyem sel'skogo khozyaystva" [Automated Control System of Production and Technical Supply of Agriculture], Nal'chik, El'brus, 1981, 256 pages.

9. Nikitin, A.I., "Obshchesistemnoye programmnoye obespecheniye sistem real'nogo vremeni" [System Software of Real-Time Systems],Kiev, Naukova dumka, 1980, 136 pages.

10. Yakubaytis E.D., "Architektura vychislitel'nykh setey" [The Architecture of Computer Networks], M, Statistika, 1980, 280 pages.

11. Baytser B., "Arkhitektura vychislitel'nykh kompleksov" [The Architecture of Computer Complexes], M, Mir, 1974, Vol 2, 546 pages.

9875
CSO: 1863/196

UDC 681.31

NEW ASPECTS OF APPROACH TO RESOLUTION OF CERTAIN PROBLEMS OF COMPUTER ENGINEERING UNDER PRESENT CONDITIONS

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 2, Mar-Apr 81 (manuscript received 12 Jun 80) pp 32-37

[Paper by Ye.I. Bryukhovich]

[Text] The decisions comprising the basis for modern computer equipment, which have already become classical, have also come to be considered optimal although there has never been any mathematical formulation of the problem of optimizing many of them. To be numbered among such decisions are primarily the binary numeration system, as well as the method of machine representation of binary numbers, the technique of automatic modulo $m = 2^k - 1$ checking of calculations (in practice, for values of $k = 2$), etc. It is well known that the decisive part in their selection was played by the fundamental proposition that the number of numerical digits M_p, contained in n_p p-nary number positions, which is equal to pn_p, assumes a value when $p = 2$ which is only slightly inferior to the minimum (when $p = 3$). This provided the basis for the assumption that the equipment costs of the machine representation of binary numbers and the time needed to perform operations on them are also minimal. An additional argument in favor of binary numeration is the fact that elements having p stable states, which are needed to represent the numbers, become the simplest when $p = 2$ [1, 2].

It has become obvious in recent years because of a number of studies which have been made that the assertion of the maximum effectiveness of binary information coding in providing for computer speed is erroneous, if it is not accompanied by stipulations concerning limitations regarding hardware costs (for example, [3-5]). The question of the machine representation of nonbinary numbers is also a similar one: it is known that each of these numbers, by virtue of their own binary nature (it is either utilized or not utilized in writing the given number) can be represented by a single component with two stable states [6]. In this way the same simplicity of the representation is assured as in the case of binary numeration, but in this instance, considerably higher equipment expenditures are required, since p bistable components are needed for the representation of p numbers in each digit.

Thus, that basis on which the decisions were made as regards the information fundamentals even for the first generation computers to a considerable extent is no longer in line with the actual state of affairs if one disregards the practical minimum in hardware expenses which is provided by binary numeration.

It has become known recently that the method of machine representation of binary numbers by means of one component with two stable states was the reason for the occurrence and existence of the problem of automated checking of computations in a computer - one of the most complex and urgent problems in modern computer engineering [6]. Considered briefly, the essence of this question reduces to the following.

1. Positional notations with an arbitrary base, $p \geq 2$, possess a natural checkability, which significantly exceeds that which can be provided through the incorporation of additional digits in the number for modulo checking. The level of this checkability is sufficient for solving the problem of automated computer monitoring.

2. Numeration systems having the same base impart natural redundancy functions for error detection to switching functions, expressed in conjunctive or disjunctive normal form. As a consequence of this, the checking system functions are organically tied to the nominal functions of the set of assemblies around which the computer is designed, and the necessity of an independent test system in the classical "tested unit--test system" configuration is eliminated. This removes one of the most difficult barriers to the ways of solving the problem. To resolve this problem, it is necessary to have an adequate representation of these properties of positional notation systems, which, as has been established, is provided by the technique described above for representing each of p numbers with a single bistable component. Thus, the potential capabilities of positional numeration systems both in resolving the problem of checking and in boosting computer speed require changes in the existing information bases of computers, something which entails a considerable increase in equipment expenditures. However, the traditional view of the existing information principles of computers (notation and the manner in which figures are represented) both for those which to the greatest extent satisfy all of the requirements of theory and practice of electronic computer design, do not make it possible to utilize these capabilities to significantly enhance the performance of present and future computers.

The main obstacle here remains the undoubted correspondence of binary numeration to the practical minimum of hardware expenditures for computer construction. In terms of canonical representations, this presupposes the attaining of the greatest effect in cost and reliability characteristics, as well as in power consumption, etc. However, the actual relationship between hardware expenditures and this effect no longer conforms to traditional views under present conditions.

The substantiation of this assertion is the goal of this paper.

For this, we shall turn to one of the major indicators of the national economic efficiency of any industrial product: the economic efficiency, ε, defined as [7]:

$$\varepsilon = A/W$$

where A is the economic impact of the introduction of the product, while W is the national economic expenditures related to producing the impact A. The

quantity W is composed of three main groups of expenditures: nonproduction expenditures (W_1), related to the development of the new computer model and its software, production expenditures (W_2) and those related to the introduction and operation of the computer (W_3).

The quantity W_1 is composed of the sum of all of the expenditures related to the performance of the scientific research and prototype design work involving the creation of the new computer model and expenditures of a similar nature related to the creation of its software (MO):

$$W_1 = W_{BM} + W_{MO}$$

[BM = computer hardware]. These expenditures are associated with the cost of each N computers, provided by the program for the production of the given model of machines.

In terms of structure, the most complex expenditures are W_2, which comprise the total production cost of the computers. They are composed of the plant production cost and the expenditures related to placing the new computer model in production (working out the design documentation and the production process, performing tests, developing and manufacturing test and quality control as well as other nonstandard equipment), designing the packaging, as well as packing and shipping the finished product, etc. The bulk of the expenditures in this group is the plant production cost, C_1, for the computer:

$$C_1 = C_м + C_{зп} + C_{оф}/N + C_т + C_{пр}, \quad (1)$$

where C_M is the cost of the materials, purchases semi-finished products and other products (including electrical parts which comprise the component base for the model)*; $C_{ЗП}$ are the wages for all plant workers, referenced per unit of manufactured product (per computer)**; $C_{ОФ}$ is the cost of the fixed capital; C_T is the cost of fuel and energy for production process purposes, and $C_{ПР}$ are other expenses.

The W_3 expenses are composed of the one-time capital investments C_{KB} (expenses of transporting the computer to the operating site, the cost of the production space occupied by the computer, expenditures for changing the system in which the computer is incorporated) and the costs $C_Э$ of computer operation.

Thus:

$$W = W_{вм}/N + W_{мо}/N + C_м + C_{зп} + C_{оф}/N + C_т + C_{пр} + C_{кв} + C_э. \quad (2)$$

*The cost of the materials and semi-finished products is determined in separate articles.

**The quantity $C_{ЗП}$ is composed of the basic and supplemental wages for production workers, as well as deductions for social insurance from the wages of production workers, and in fact, all of these components are calculated in separate articles.

We shall make use of this expression to retrospectively estimate those decisions which have been taken as the basis for the operation of computers of all generations as well as to analyze the conditions which governed the making of these decisions.

The basic design principles of computers not affected by the change in generations (the program stored in the memory and the information principles of the machines) were embodied even in the machines at the outset of the era of electronic digital computer engineering. These computers (for examples, the MESM [small electronic computer] - the first computer in our country and in continental Europe) represented for that time unique creations of man not only in terms of the unprecendented computational capacity, but also the areas and possibilities of their application. All of the electronic components with which the electronics of these machines were constructed were incorporated in assemblies as constituent elements (standard logic gates, memory elements, etc.), the products list of which was more than three orders of magnitude less than the total number of vacuum tube devices used in the machine. Such a high applicability of standard assemblies also created unprecedented capabilities for the reproduction of computers on an industrial basis. However, the substantial differences in the production technology of vacuum tube devices and electronic components did not allow the fabrication of these components and standard assemblies in a single technological process with a high level of mechanization and automation of labor. To a considerable extent, the manufacturing of the standard assemblies permitted the use of only manual labor, because of which computers could not produced in large series, not to mention mass production. For this reason, such a powerful means of reducing the costs W as increasing N, which is attainable only with large series and mass production, could not be fully utilized at that time.

The share of each component of the right side of expression (2) remained high, but not just for this reason alone. The process of developing the computers and their software necessitated large expenditures of manual labor, since due to the lack of the computers themselves in the requisite quantity and assortment as well as the lack of a developed computer aided design theory, it was not possible to use them in all stages of the scientific research and prototype design work. As a consequence of this, the W_{BM} and W_{MO} expenditures were high. An analysis of the other components of the right side of expression (2) shows that the single effective means of reducing the quantity W at that time was reducing the hardware expenditures. It led not only to an obvious reduction in the quantity C_M, but also to a reduction in the total volume of work involved in manufacturing the machines, to a reduction in the quantity $C_{ЭЛ}$, and to some extent, to a reduction in C_T. The nature of the influence of the equipment expenditures on the C_{KB} and $C_Э$ components is similar. This is due to several factors.

First of all, the operating economy of vacuum tube devices as regards power consumption was extremely poor. Thus, the 6N8S, 6N9S and 6Kh6S vacuum tubes with which the MESM was constructed consume about four watts of power each in just the filament circuit alone, and taking into account the total number of them in the computer, they consumed more than 16 KW. Because of this, the cost of solving problems on such machines was rather high. Another negative consequence of the low operating economy of vacuum tubes was the poor computer utilization factor during a 24 hour period. This was related to the fact that practically all of

the power consumed by the machine was radiated into the surrounding space in the form of heat. This created a considerable local temperature rise in the machine over the ambient temperature. If this ambient temperature rose above a certain level (for example, during hot summer days), the operability of the machine fell practically to zero because of overheating. The only way of eliminating or even slightly cushioning the impact of this deficiency, given the component base of that time, was reducing the total number of power consuming devices, i.e., the total number of vacuum tubes.

Secondly, tube reliability was extremely poor. According to data in the literature [8], the mean time between failures of the MESM computer was a few hours. Because of this, the costs of servicing the computer were extremely high. A reduction in these costs was related primarily to a reduction in the total number of tubes in the computer.

Thirdly, the physical dimensions of the tubes (even the so-called baseless and miniature series) were relatively great. Along with the necessity of spacing the components in the machine to provide for efficient heat removal, they were also responsible for the considerable physical size of the machines, and as a consequence, of the rooms in which the machines were operated. This held up the development of computer equipment to a definite extent, and consequently, did not create an objective demand for increasing N, since computers of this size could be utilized only in limited sectors of the national economy. Another consequence was the high value of C_{KB}, which could also be reduced only by decreasing the number of tubes in the computers.

It follows from what has been presented here that for first generation computers, progress in the development, refinement and wide scale introduction of computer hardware was related primarily to reducing equipment costs. The cost, reliability, power consumption and physical dimensions of the machines were then the major characteristics of computers, and for this reason, the effectiveness of particular design solutions which were adopted in the development of the computers was estimated primarily in terms of their impact on an improvement in these characteristics. Since as we have seen, the only and radical means of simultaneously improving these characteristics remained the reducing of equipment expenditures, the criterion of minimal equipment expenditures justifiedly became the major criterion at that time for evaluating the effectiveness of all of the design decisions comprising the basis of computer operation. In particular, a radical consequence of this was the adoption of binary numeration as the information principle of the computers and the method of computer representation of numbers, which is now also being used in practically all computers presently in service and those being developed. Thus, more than 30 years after the main principles of computer design were determined [9], it remains an undoubtable fact for us that the choice of binary numeration and binary representation of numbers by means of a single element with two stable states, which was made for the first computers, was completely justified then.

However, the data of paper [8] provide evidence that this measure alone was nonetheless insufficient to radically improve computer performance, and thereby resolve the problem of significantly curtailing the national economic costs W and bring about the extensive introduction of computer equipment in all spheres of human activity.

The appearance of semiconductor hardware, which gave birth to the next generation of computers, played an objective role as a new factor in reducing the expenditures W. The miniature size, rather high reliability and operating economy in terms of power consumption were responsible for reducing the costs C_{KB} and $C_{э}$. Because of this, the range of practical applications of computer hardware was substantially expanded, something which in turn was responsible for a certain increase in N. However, the major importance of semiconductor technology consisted in the fact that it created the prerequisites for developing the technology (called integrated circuit technology) of fabricating entire fragments of the electronics of a computer in a single production process. The invention of integrated circuit technology was a true revolution in the nature of computer production and had a multiplicity of fundamental consequences which exerted a decisive influence on the further development of computer equipment. We have singled out the following from among them:

1. Integrated circuit technology made it possible to automate the most labor intensive portion of the production cycle of computer manufacture: the production of the component base. This created the basis for the large series production of computers.

On the other hand, integrated circuit technology made it possible to sharply extend the range of computer equipment uses. The objective basis for this was the following factors. First of all, the reliability of the component base improved sharply: the guaranteed mean time between failures was now already no less than 10^4 hours [10]. This significantly reduced the specific share of the quantity $C_{э}$ in the overall expenditures W. Secondly, the power consumed by the IC's was substantially reduced: as compared to the power consumed by first generation computer components, the level of reduction amounted to three orders of magnitude [10]. Because of this, the specific share of $C_{э}$ in the expenses W was reduced even more. Thirdly, integrated circuit technology significantly reduced the physical size of the machines, because of which the C_{KB} costs were also reduced. In addition to this, broad possibilities were opened up to using computers where assuring minimum size, weight and power consumption was simply essential.

This conformity between production capabilities and the demand for computer use created qualitatively new conditions for assuring a minimum of W. Equipment expenditures have now ceased to be the single factor in the reduction of the value of W, while increasing N has come to play the highest ranking role in this regard. As a consequence of this, it has become possible to increase equipment expenditures in computers to improve the economic impact A and to compensate for the unavoidable increase in W in this case by reducing other components of W which depend on A.

However, these capabilities are nonetheless limited at that level of integration which was achieved for series produced integrated circuits. The cost of all of the integrated circuits used in a computer is determined by the expression:

$$C_{IC} = C_{ис} = \sum_{k=1}^{n} \sum_{l=1}^{m} C_{иск,l} = \sum_{k=1}^{n} \sum_{l=1}^{m} (C_{мk} + C_{зпk} + \quad (3)$$

$$+ C_{оф,ис}/nmN + W_{иск}/nmN + C_{1k} + C_{прk})_l,$$

where $W_{ис,k}$ is the cost of developing integrated circuits of the k-th type; n and m are the number of types and the number of k-type integrated circuits used in the given computer, respectively; C_{Mk}, $C_{эпk}$, $C_{гk}$, $C_{прk}$ and $C_{оф,ис}$ are the symbols used above, but applied to IC production.

It follows from expression (3) that a further reduction in the value of W is now not related to reducing the equipment costs in the sense used in the time before integrated circuits, but rather to decreasing the quantities n and m.

2. The importance of integrated circuit technology goes beyond the scope of automating the production of the component base. The specific fundamental feature of it consists in the fact that it offers the capability of increasing the level of integration of integrated circuits up to the ultimate level at which the entire electronics portion of a computer will be on a single chip. This completely resolves the major problem of reducing W, which is related to decreasing n and m. However, it is important here that the realization of this capability is not accompanied by a change in the production cycle for the manufacture of integrated circuits. This leads to the fact that when the level of integration rises, the microcircuit cost changes only as a function of the change in just the single quantity $W_{ис}/N$ [W_{IC}/N] and in terms of this, is dependent on the number of gate devices placed on the chip. The other components of $C_{ис}$ [C_{IC}] do not depend on the number of these gates. Consequently, it can be concluded that an increase in equipment costs for the purpose of boosting the economic impact A leads to a change in the cost of a LSI computer only by the amount $\Delta W_{LSI} = (W_{LSI} - W_{IC})/N$, where W_{LSI} are the expenditures related to the development of the LSI computer in a single chip design.

Consequently, when the ultimate level of microcircuit integration is achieved:

$$C_{бис} = C_м + C_{зп} + C_{оф}/N + W_{бис}/N + C_т + C_{пр},$$

[$C_{бис} = C_{LSI}$] where C_{LSI} and W_{LSI} are the costs of the microcircuit and its development with a single chip design for the entire electronics portion of the computer (LSI computer).

A level of integration has already been attained which is adequate for the single chip fabrication of the major components of a computer: the processor and the memory. In fact, several types of domestic single chip microprocessors have been developed at the present time and are already being series produced, and the time is not far off when all of the electronics of a computer will be produced on a single chip.

There follows from this the extremely important conclusion that integrated circuit technology has practically resolved all of those problems which arose in the design of the first computers, and now the factor of the economic impact of equipment costs has lost the importance which it had in all previous generations of computers. It also follows from this that state of the art integrated circuit technology has now already eliminated the basis on which the application of binary numeration and the technique of the machine representation of bits by an element having two stable states were justified.

It can also be concluded based on what has been said here that integrated circuit technology, having achieved an appropriate level of component integration, will make it possible to sharply boost the economic efficiency of computers by increasing the economic impact A for the case of constant expenditures W, if this impact is achieved through an increase in hardware costs.

The following question takes on special importance in light of these conclusions: whether the information principles of modern computers are appropriate for the situation which has come about or whether are still factors which prohibit changing these principles for the purpose of increasing the economic impact A? In order to answer this question, we shall turn to problems of a design and production process nature, which by virtue of the specific features of integrated circuit technology, are brought to the fore by the major trend in its development: the increase in the level of microcircuit integration.

Besides purely technological problems, the major factors retarding progress in resolving the problem of increasing the level of integration are the permissible power consumption of the microcircuit and the limited number of leads for standard IC packages.

The level of power consumption by an integrated circuit is determined by the number of elements which consume energy, i.e., in the final analysis by the total number of digits $M_p = pn_p$ in a number.

Consequently, insofar as the power consumption of a microcircuit is concerned, of all of the other numeration systems, it is specifically binary notation which meets the requirements of integrated circuit technology to the greatest extent. As early as several years ago, this was also actually the case. However, recent developments in the field of structures with comparatively low power consumption have made this principle extremely relative. One of the most important ones here is the development of CMOS structures, a specific feature is the practically zero power consumption in their static operating mode and relatively large consumption in the switching mode [11]. This brings to mind the necessity of a machine representation of information which would provide a minimum number of element switching operations. To what extent do the information principles underlying modern computers conform to these new conditions?

In order to answer this question, we shall turn to the first method described above for the machine representation of numbers in a numeration system having a bases $p \geq 2$. When this method is used, each number of each digit is represented by means of one element with two stable states. Consequently, to represent all of the numbers of each digit, p such elements are needed, and to represent the entire number, pn_p such elements are required. Since in each given point in time there is always only one number written in one digit of a number in positional notation, the number of elements which actuate when writing a p-place number into a memory register is n_p, while $(p - 1)n_p$ register elements in this case are in the static mode. For this reason, $\beta = n_2/n_p > 1$ for all $p > 2$. It follows from this that binary numeration, if unitary numeration is not taken into account, is the most ineffective when CMOS structures are used.

It also follows from this that since $n_p = n_2/\log_2 p$ and $\beta = \log_2 p$, there is a substantial margin for increasing the economic impact of saving energy by choosing a high value of p. This is extremely important for CMOS structures, which are inferior in terms of speed to bipolar structures, since a higher value of p also provides for greater computer speed [3, 4]. Limitations here are related only to the attainable level of integration, which depends on the resolving power of the production process equipment. Such a margin is naturally lacking when binary numeration is used.

What has been presented here also makes it possible to draw the more general conclusion: the quantity of equipment, in the sense employed in the first generations of computers, is not only not to be reduced, but to be increased if this increase leads to a reduction in the number of switching operations of the elements in the integrated circuit, since this increase in practice does not lead to an increase in integrated circuit cost. Bipolar structures in which the components have switched collector impedances also exhibit a similar effect. The power used in switching such elements is several times greater than the power consumed by the elements in the static mode [11].

The question of a limitation as regards the permissible number of leads in a package is not problematical, and is to a certain extent, a purely technical issue. First of all, considering the present phase in the development of integrated circuit technology, it is impossible to preclude the possibility of changing the standards for IC packages for the purpose of increasing the number of leads in them. Secondly, in this same phase of monolithic execution of processors and memories, the conversion of the positional representation of p-place numbers in one of the forms characteristic of well-known elements with multiple stable states can also be employed without difficulty (including also a binary code). The mutual conversion of these forms and the positional code at the input and output of the circuits will make it possible to incorporate packages in that products list which is now being used in series production. For this reason, limitations as regards the permissible number [of leads] in a package may be disregarded.

What has been said here applies to the governing trends in the development of computer hardware and makes it possible to draw the following conclusion.

The start of this stage in the development of integrated circuit technology has been clearly marked out at the present time, when an increase in hardware costs within reasonable limits to achieve the requisite economic impact is in practice not accompanied by an increase in the overall economic outlays. A consequence of this is that conditions are created, especially with a single chip design of the entire electronic portion of a computer, where increasing the hardware costs becomes not only possible, but also expedient if it is due to the necessity of increasing the economic impact A.

Under these conditions, binary numeration and computer number representation, which are traditionally used in electronic computers, become significantly less effective than numeration systems with a larger base and positional notation for the machine for the machine representation of their numbers, even for the entire set of criteria, on which the maximum effectiveness of the traditional information

principles of computers was acknowledged in the earlier period of development of computer hardware.

For this reason, there are significant objective factors which would impede making a decision to change the information fundamentals of computers in favor of non-binary numeration systems with positional notation for the machine representation of the numbers. This will allow maximum utilization of the potential capabilities of positional notation systems in the resolution of traditional problems of the automated testing of computers as well as computer speed and power consumption.

BIBLIOGRAPHY

1. Kartsev M.A., "Arifmetika tsifrovykh mashin" ["The Arithmetic of Digital Machines"], Nauka Publishers, 1969, 576 pp.

2. Papernov A.A., "Logicheskiye osnovy tsifrovykh mashin i programmirovaniya" ["Logic Fundamentals of Digital Computers and Programming"], Moscow, Nauka Publishers, 1968, 592 pp.

3. Bryukhovich Ye.I., "Vliyaniye pozitsionnogo predstavleniya dvoichnykh i nedvoichnykh tsifr i velichiny osnovaniya schisleniya na bystrodeystviye EVM" ["The Influence of the Positional Representation of Binary and Nonbinary Digits and the Size of the Radix on Computer Speed"], USiM [CONTROL SYSTEMS AND MACHINES], 1978, No 2, pp 67-72.

4. Bryukhovich Ye.I., "O vliyanii pozitsionnogo predstavleniya i osnovaniya schisleniya na vremya vypolneniya umnozheniya v TsVM" ["On the Influence of the Positional Representation and Radix on Multiplication Execution Time in Digital Computers"], USiM, 1979, No 2, pp 65-69.

5. Ivas'kiv Yu.L., "Printsipy postroyeniya mnogoznachnykh fizicheskikh skhem" ["Design Principles of Multidigit Physical Schemes"], Kiev, Naukovo Dumka Publishers, 1971, 316 pp.

6. Bryukhovich Ye.I., ["O Probleme avtomaticheskogo kontrolya v EVM i kontrolesposobnosti pozitsionnykh schisleniy"] ["On the Problem of Automatic Monitoring in Computers and the Checking Capability of Positional Notation Systems"], USiM, 1977, No 4, pp 71-75.

7. Konson A.S., "Ekonomika priborostroyeniya" ["The Economics of Instrument Making"], Moscow, Vysshaya Shkola Publishers, 1970, 342 pp.

8. Abalyshnikova L.M., Parkhomenko I.T., "Nekotoryye vyvody po ekspluatatsii radiolamp v MESM" ["Some Conclusions from the Operation of Vacuum Tubes in the MESM Small Electronic Computer"], in the book, "Voprosy tekhniki bystrodeystvuyushchikh schetnykh mashin" ["Questions in High Speed Computer Engineering"], Kiev, UkrSSR Academy of Sciences, 1955, No 1, pp 29-43.

9. "Proyektirovaniye sverkhbystrodeystvuyushchikh sistem. Kompleks 'Stretch'" ["Ultrahigh Speed System Design. The 'Stretch' Complex"], Edited by V. Bukhgol'ts,Moscow, Mir Publishers, 1965, 332 pp.

10. "Spravochnik po integral'noy tekhnologii mikroskhem" ["Handbook on Integrated Circuit Technology"], Edited by B.V. Tarabrin, Moscow, Energiya Publishers, 1977, 584 pp.

11. "Spravochnik po tsifrovoy vychislitel'noy tekhnike" ["Digital Computer Engineering Handbook"], Edited by B.N. Malinovskogo, Kiev, Tekhnika Publishers, 1974, 512 pp.

8225
CSO: 1863/171

UDC 519.685

THE K580/LGU CROSS SYSTEM

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 (signed to press 16 Jun 82, after revision 15 Sep 82) pp 19-21

[Article by Sergey Nikolayevich Baranov, Vyacheslav Alekseyevich Kirillin, Andrey Alekseyevich Klubovich, Nikolay Romanovich Nozdrunov and Aleksandr Leonidovich Sakharov, "The K580/LGU Cross System", in the section "Microprocessor Technology and Its Application"]

[Text] The K580/LGU cross system was created by the team of authors at the Computer Center of Leningrad University as a tool for developing general purpose hardware for devices based on K580 microprocessors. The system's computer was the YeS, with a memory size of at least 256 Kbytes, operating in the YeS OS medium.

The cross system is oriented toward users of rather high skill, and performs the following stages in development of the program product:

independent translation of modules of the program developed in a certain general context;

combination of previously translated modules into a single load one, and its adjustment to the specified main memory addresses;

translation and debugging of modules simulating a situation that is external relative to the program product developed; and

interpretation and debugging of the previously obtained load module with a specified simulation of the external situation.

Each stage is performed as a separate job or job step under control of the YeS OS in the batch mode.

Programming of the component modules of the program product developed is done in the high-level ASSOL/M language and K580/LGU macroassembler. The assembler of the INTEL firm [1] is also allowed, for combination with previously created software.

The ASSOL/M language is a variation of the ASSOL system programming language [2], and is designed for the K580 eight-bit microprocessor. It is a high-level language with the convenient program and data representation common to such languages and

reliable programming. At the same time, it is intended for a specific microprocessor, thus enabling a program whose efficiency is considerably better than that of programs obtained by translators from widely used high-level languages such as PL/M, PASCAL, and SI, and generally as good as program modules written manually in the assembler. Specific language constructions enable access to all the microprocessor's hardware.

The language has widespread structures of control and data, enabling program development along the lines of structural and module programming. The ASSOL/M is a high-level macrogenerator, and does not load the working program with a standard environment, thus ensuring its high efficiency.

The cross system contains a translator, converting the text from the ASSOL/M language to the assembler. Both the K580/LGU macroassembler and the INTEL assembler can be used. The text obtained can be used as any other text in the assembler, meaning that it can be corrected manually (for instance, to enhance program efficiency). It can also be used as part of a certain module text, as an input text for the macrogenerator, or for immediate translation by the assembler.

The K580/LGU assembler uses the same machine instruction mnemonics as the INTEL assembler [1], but differs in the capabilities of the assembler instructions (pseudoinstructions) and the possibility of using Russian letters in the identifiers and string constants. Overall, this language is more advanced than the INTEL assembler.

The cross system contains a translator from the K580/LGU assembler, translating the text in this language to a relocatable object code. To automate the carry of programs operating in the INTEL assembler, there is a converter program, converting the text from this language to a text in the K580/LGU assembler.

The macrotools of the K580/LGU assembler are the "superstructure" over the assembler itself, using as the macrogenerator a standard OS YeS macroassembler as the universal macrogenerator. Macrodefinitions are written in the syntax of the OS YeS macroassembler, but generation of assembler instructions is specified not as individual operators with possible macrosubstitutions, but in the field of the operand of PUNCH operators or in operators directly following the REPRO operators. The syntax of the macroinstruction operators in the assembler text subject to macrogeneration is the same as in the YeS computer macroassembler.

The cross system contains a preprocessor, converting the input text in the K580/LGU macroassembler to a text meeting the requirements of the OS YeS assembler macrogenerator. After the text received is processed, this macrogenerator is used to obtain a text in a "clean" assembler, without macrooperators. The cross system's macrogenerator thus consists of a preprocessor program and a standard macrogenerator of the OS YeS assembler. When specifying a special parameter, this macrogenerator can process the input text in the INTEL macroassembler with the same agreements on the method of recording macrooperators and macrodefinitions.

The cross system contains a link editor, enabling combination of separately translated object modules into one load one, with its adjustment to the specified addresses of the microprocessor's main memory. The basic functions of the link

editor are to allow external references with the possibility of automatic search of modules and control of location of the modules switched in at the code and data segment level. The load module obtained can be used for interpretation, for direct loading into the microprocessor followed by execution (when outputting onto punch tape), for putting a program into the ROM, and so forth.

The cross system contains the MONITOR subsystem, which is an independent set of general purpose tools for writing and debugging multimodule programs in the YeS computer assembler. Within the cross system, MONITOR is used to create modules simulating a situation external relative to the program developed (operating system functions, input-output ports, interrupt system, etc.). The MONITOR subsystem consists of a load module ready for execution and a library of macrodefinitions (in the Os YeS macroassembler). It performs interface between the OS YeS and the program for the YeS computer, translated using its macroinstructions and considering a certain context, and provides a specific link mechanism between the component modules of this program (call with transmission of parameters and results and processing of exceptions). Tools are also provided for access to OS YeS text files, editing and output of data to the standard output file, main memory allocation, and debugging of a given program.

The cross system's interpretor enables use of a binary code, loaded ahead of time into the main memory of the K580 virtual microprocessor, counting of the execution time, detection of errors and ordered events, and program interruption. The interpretor receives a previously created virtual microprocessor with a loaded binary code as the parameter for its operation. Several virtual microprocessors can be worked with when only one interpretor copy is used.

For a given virtual microprocessor, the interpretation tools enable events to be ordered and detected during operation. Such events can be references to specific main memory addresses and microprocessor registers, control transmissions to specific addresses, and expiration of a given amount of time. The interpretor also enables recognition of exceptional events related to errors in the program interpreted or execution of instructions affecting its environment (IN, OUT, HLT, and so forth).

When an ordered or exceptional event occurs, interpretation ceases and control is switched to the module that started it. This module processes the situation, during which it can again refer to the interpretor to define more clearly the event detected, introduce changes in the microprocessor status and renew interpretation from the interrupted point or any other.

The cross system contains a debugging subsystem, which is a simple tool for employing the interpretor's capabilities. It provides convenient access to all its tools using a special language (based on the OS YeS macroassembler), and provides the user with the following capabilities:

creation of a virtual microprocessor;

loading a program for interpretation into this virtual microprocessor's main memory;

ordering events and reactions to them; and

starting interpretation of the loaded program, with allowance for the events ordered.

The debugging language contains programming language elements for specifying uncomplicated reactions to the ordered events directly in this language. With more complicated processing of events, the corresponding module should be prepared beforehand using the tools of the MONITOR subsystem.

The cross system described has been in use at the VTs LGU [Leningrad State University Computer Center] since 1981. During this time, it has been used to create a data text editing program on a display and several application programs for the YeS7970 display complex.

The experience of its use shows that it enhances programmer productivity and enables production of high-quality, reliable software. For instance, the volume of the text editing program mentioned was 6.5 Kbytes (about 2500 lines of text in ASSOL/M); three man-months were spent on its generation.

BIBLIOGRAPHY

1. INTEL-8080/8085. Assembly Language Programming Manual. INTEL Corp., 1977, 160 pages (Order Number 9800301B).

2. Tseytin G.S., "The ASSOL System Programming Language", in "Perspektivy sistemnogo i teoreticheskogo programmirovaniya" [Future of System and Theoretical Programming], Novosibirsk, VTs SO AN SSSR, 1979, pp 133-139.

9875
CSO: 1863/196

UDC 681.3.06.51

METHOD OF RUNNING PROGRAMS ON COMPUTERS WITH SIMILAR ARCHITECTURE

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 (signed to press 23 Jun 82) pp 34-37

[Article by Leonid Konstantinovich Bigday, Aleksandr Zinov'yevich Genelis, Aleksandr Genchev Ivanov and Igor' Yakovlevich Landau, "A Method of Running Programs on Computers With Similar Architecture", in the section "General Software of Control Systems"]

[Excerpts] This work has been performed within the SETL project, whose goal was to realize a superhigh-level language on several computer types. The project includes investigation of the problem of software portability and methods of translating it to other machines, notably those with a similar architecture. The result of the work has been the development and realization of a method of running programs on computers with a similar architecture, or on different operating systems of one computer. The method does not require the participation of the source machine from which the translation takes place.

The author of the SETL language is New York University professor J.T. Schwartz [1], who directed the first realization of the language [2]. The basis of SETL is operations on sets, enabling its effective use to describe and check complex algorithms [3-6]. In view of the complexity of realizing compilers of such level languages, the LITTLE instrument language was developed for SETL [7-8], and then a programming system based on it. The SETL compiler and its executive system were written in LITTLE, which has proven itself well as an instrumental tool and has been used in developing the BALM system for programming the processing of lists and the operating system for the artificial intelligence language ARTSPEAK [9].

LITTLE has the necessary properties of a system programming language (such as efficiency, modularity, and the presence of bit and symbolic string processing tools), while providing mobility when translating programs to other machines [10]. Translation thus enables LITTLE to solve the problem of running systems based on it, including SETL. LITTLE was initially realized on the CDC-6600, then translated to the IBM/370, PDP-11, VAX-11/780 and others.

This article describes the authors' translation (and interpretation) of the LITTLE programming system from the IBM/370 with the OS/MVT operating system to the M4030 with the DOS ASVT operating system.

Conclusion. The LITTLE compiler controlled by the DOS ASVT requires 210K bytes of memory and operates at 400-500 lines of source text per minute on the M4030-1. It satisfies system conventions on communication used in the DOS ASVT. There are special macroinstructions for users who wish to connect the LITTLE program with the assembler.

The work to translate and interpret the LITTLE system required writing and debugging about 6,000 program lines in LITTLE and the assembler, and took three man-years, which is 4 to 5 times less than the cost of developing the initial system for both indices.

There is presently a tendency to develop computer series with a similar architecture, and give them different operating systems. As a rule, the software for different operating systems is generated independently, and cannot be translated to other operating systems without a major loss of efficiency.

The method for translating object modules proposed by the authors can be used to translate compilers and application programs in the YeS and SM series computers and foreign computer series compatible with them.

The authors thank Professor J.T. Schwartz and his assistants for providing materials on the SETL-LITTLE system and discussing joint work at a seminar in Moscow.

BIBLIOGRAPHY

1. Schwartz J.T., "Abstract Algorithms and a Set Theoretic Language for Their Expression", New York, CIMS, 1968, 290 pages.

2. Schwartz J.T., "On Programming: An Interim Report on the SETL Project", New York, CIMS, 1975, Parts 1, 2, 676 pages.

3. Chernobrod, "A Review of Basic Concepts of the SETL Programming Language", in "Sistemnoye i teoreticheskoye programmirovaniye" [System and Theoretical Programming], Novosibirsk, SO AN SSSR, 1972, pp 236-277.

4. Ivanov A.G., "The Set Theoretical Language SETL", TRUDY INEUM, 1975, Vol 48, pp 41-51.

5. Levin D.Ya., "SETL: A Very High Level Programming Language", PROGRAMMIROVANIYE, 1976, No 5, pp 3-9.

6. Ivanov A.G., ""Downward" Programming Using a Superhigh Level Language", TRUDY INEUM, 1978, Vol 70, pp 47-53.

7. Cocke J., Schwartz J.T., "Programming Languages and Their Compilers", New York, CIMS, 1970, 768 pages.

8. Shields D., "Guide to the LITTLE Language", New York, CIMS, 1974, 98 pages (LITTLE Newsletter 33).

9. Shields D., "Proposals for the Next Stage LITTLE Development", New York, 1973, 6 pages (LITTLE Newsletter 24).

10. Ivanov A.G., "LITTLE: A Language Realizing Mobile Programming Systems", TRUDY INEUM, 1980, Vol 80, pp 33-36.

11. Lecarme O., Peyrolle-Thomas M.-C., "Self-Compiling Compilers: An Appraisal of Their Implementation and Portability", Software; Practice and Experience, 1978, Vol 8, pp 149-170.

12. Yushchenko Ye.L., Pogorelyy S.D., "Program Compatibility of Computers and Methods of Realizing It", KIBERNETIKA, 1975, No 1, pp 54-63.

13. Early J., Sturgis N., "A Formalism for Translation Interactions", Comm. ACM, 1970, Vol 13, No 10, pp 607-617.

14. Geneles A.Z., Ivanov A.G., "Realizing Interface From the DOS-2 ASVT in the LITTLE Compiler", TRUDY INEUM, Vol 87, 1981, pp 24-29.

9875
CSO: 1863/196

UDC 681.32

METHOD OF CREATING A FAMILY OF DEDICATED REAL-TIME OPERATING SYSTEMS

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 (signed to press 3 Aug 82, after revision 10 Dec 82) pp 47-52

[Article by Kalev Imarevich Kyaeramees and Leo Leovich Mytus, "A Method of Creating a Family of Dedicated Real-Time Operating Systems", in the section "General Software of Control Systems"]

[Excerpts] This article considers the possibility of changing the properties of a virtual machine by using the nucleus of an operating system, drivers of the external devices and schedules to organize coordinated multiprogram operation of application software. By operating system is meant only the control program. System programs (such as compilers, link editors, text editors and debugging aids) belong to application programs.

The nucleus of the operating system contains primitive instructions, which expand the set of processor instructions. Examples of primitives are tools for synchronizing parallel executed programs, tools for transmitting messages, for reference to the resource allocation system, for managing time services, etc. Tools for using time services and for message synchronization and transmission are particularly well developed in real-time operating systems.

In this article, the discussion uses the example of the OS RV [real-time operating system] MEDEX, used to dedicate the YeS1010 computer for various applications (such as carbamide production process control, on-board systems of geophysical data collection and processing, in the cross system of programming microprocessors for simultaneous operation of four programmers, and others). Choosing the nucleus primitives, external device drivers and schedulers makes it possible to consider the specific requirements of different users, as well as the varying demands on system survivability and operating reliability.

The focus is on methodological aspects, not on describing specific applications.

This article is based on the modular and aggregate methodology of OS RV design [1-3], but differs in its more detailed structuring of the OS RV. The authors' goal is to define the practically important points of OS RV specialization and develop tools for effectively using the capabilities of such OS RV's when building application systems. In this sense, this article's methods are applicable in the practical realization of the results of systems for designing software of built-in systems [4-6].

Conclusion. In real-time systems, it is often preferable to use a dedicated virtual machine in place of a dedicated computer: to build a dedicated OS RV. With current module libraries and generation tools, it takes 1-6 man-months to make a new member of the OS RV family, depending on the required degree of specialization.

It is also sometimes advisable to solve the opposite problem. Based on physically different computers, one virtual machine is built for application programming, such as for the Elektronika-60, SM-3 and SM-4, and possibly the SM-1 and SM-2. This approach will undoubtedly save considerable resources spent on unportable application programs. Such an approach is obviously worthwhile only if the application software is written in a high-level language.

Experience demonstrates that dedication occurs basically in the following directions:

removal of superfluous services from the nucleus; i.e., minimization of the need for working memory;

addition of new services to the nucleus;

development of a new system for processing special cases, including a special protection system for power failures;

development of problem-oriented database management tools;

development of schedulers practically realizing the structure of the software application system; and

development of a hardware/software system of loading the OS RV and application software, and activating them in the specified sequence.

Using dedicated OS RV's aids in the use of a descending programming technology and allows the use of less skilled programmers to code and debug the processing programs of real-time system software.

BIBLIOGRAPHY

1. "An Aggregate System of Software of M7000 ASVT-M", A.B. Ayzenberg, V.G. Vinokurov, V.M. Kostelyanskiy et al, USiM, 1976, No 6, pp 93-98.

2. "Constructing the Nucleus of Operating Systems of an Aggregate System of Software", A.B. Ayzenberg, V.M. Smolkin, N.V. Filippov, Ye.V. Shcherbakov, USiM, 1978, No 2, pp 47-50.

3. Ayzenberg A.B., "Development of Multifunctional Operating Systems ASPO", USiM, 1980, No 5, pp 57-59.

4. Akulovskiy V.G., Kudrin V.G., Terent'yeva T.M., "A Parametric System for Designing Software of Real-Time Systems", USiM, 1981, No 1, pp 61-64.

5. Bezhanova M.M., "Analysis and Systematization of Built-In Problem-Oriented Systems", USiM, 1981, No 4, pp 113-119.

6. Motus L., Kaaramees K., "A Model Based Design of Distributed Control System Software", 4th IFAC Workshop on Distributed Computer Control Systems, 1982, Tallin, pp 1-23.

7. Parasyuk I.N., Sergiyenko I.V., "A Modular Approach to Building a Family of Packages of Application Programs", PROGRAMMIROVANIYE, 1981, No 6, pp 29-34.

8. Parnas D.L., "On the Design and Development of Program Families", Programming Methodology, Hamburg, Springer Verlag, 1978, pp 343-361.

9. "European Workshop on Industrial Computer Systems. TC-8 Up-to-Date Report", 1978, 100 pages.

10. Mytus L.L., Kyaeramees K.I., Khanson V.V., "Generation and Evolution of Automated Process Control System Software", in "Matematicheskoye i informatsionnoye obespecheniye ASUTP: Tez. dokl. vsesoyuz. konf." [Mathematical and Database Support for Automated Process Control Systems: Papers, All-Union Conference], Chernovtsy, 1977, pp 110-112.

11. Kaaramees K., Motus L., Nurmela V., "A Real-Time Executive for a Chemical Process Control System", 1st IFAC/IFIP Symposium on Software for Computer Control, Preprints, Prague, 1979, pp 1-5.

12. Computer Control of Chemical Plants: Design Experience and Prospects", M. Littover, L. Motus, O. Randma et al, IFAC 8th World Congress, Kyoto, 1981, Vol 22, pp 7-14.

9875
CSO: 1863/196

UDC 681.3.06.51

TRANSFER OF PROGRAMS FROM DOS YeS SOURCE TEXT LIBRARIES TO OS YeS LIBRARIES

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 (signed to press 10 Oct 82) pp 54-55

[Article by Aleksandr Ruvimovich Mayorskiy, "The Transfer of Programs from DOS YeS Source Text Libraries to OS YeS Libraries", in the section "General Software of Control Systems"]

[Text] The operating systems DOS YeS and OS YeS are designed to organize the computation process on the YeS computers. Although they are intended to use the same hardware, there is still a problem of the independence of the program source text from the operating system, which has not been completely solved. Moreover, no attention has been devoted to even the mechanical transfer of program source texts from the libraries of one system to another. For example, a natural requirement when moving from the DOS YeS to the OS YeS is the presence of program tools allowing modules to be placed in the OS YeS library that are output to magnetic tape by the DOS YeS program SSERV. However, there is not such a utility program in the available software.

Modifications to be made in transferred source modules, written for example in FORTRAN, are minor [1,2]. However, performing such modifications turns into additional, monotonous, time-consuming work when a large number of modules are involved. For instance, if complexes of programs operating in the DOS YeS system consisting of 200-300 modules must be loaded into the OS YeS, correcting the source texts of the program modules requires at least 100 man-hours. Moreover, the modified modules can require debugging, albeit uncomplicated.

As a rule, when translating in the OS YeS an interconnected complex of subroutines intensively using input-output operators (referring to modules written in FORTRAN), the numbers of the files in the i/o operators must be changed. In the DOS YeS, for devices of system entry, perforation and printing file numbers 1, 2 and 3 are used by default, and numbers 5, 7 and 6 for the OS YeS. [1] recommends that this inadequacy be eliminated at the execution stage, by indicating the corresponding parameters in the DD statements of the job control language.

However, in many modules statements will probably be encountered in which the file numbers are assumed by default, along with statements using the same files with the file number clearly indicated. Moreover, the numbers 5, 7 and 6 are generally used for specific purposes in the programs loaded in the DOS YeS

medium, which prevents their use in the OS YeS for its own needs (for example, to output diagnostic messages).

If in the source texts of programs operating in the DOS YeS system the file numbers 1, 2, 3 are changed to 5,7, 6 and numbers 5, 7, 6 to 1, 2, 3, respectively, in the i/o operators, then the programs obtained are suitable for use in the OS YeS medium. If the file numbers are not specified by a variable, then these modifications can be done automatically, regardless of the program logic.

After the source texts of the programs are put into the OS YeS libraries, translation and editing are performed to obtain the load modules. Since all the information needed to form job packets for translation and editing exists at this stage, it can be formed by the program and put onto a magnetic medium, such as a magnetic tape. The jobs formed are assigned to a class, ensuring their sequential execution on the specific computer.

Using the above-described program enables a complex of programs to be transferred from DOS YeS to OS YeS libraries with a high degree of reliability and at minimal machine and labor cost.

This program was developed and is in use at the Specialized Project Design Institute for Development and Introduction of Automated Systems for Equipment With Program Control (SPKTI AS).

It can also be used to transfer any source modules from DOS YeS to OS YeS libraries.

BIBLIOGRAPHY

1. "Perekhod ot DOS YeS k OS YeS" [Transition from DOS YeS to OS YeS], O.S. Ivaniko, N.M. Ivanyutina, M.P. Kotov et al, M, Statistika, 1980, 230 pages.

2. "FORTRAN YeS EVM" [FORTRAN of the YeS Computers], Z.S. Brich, D.V. Kapilevich, S.Yu. Kotik et al, M, Statistika, 1978, 284 pages.

9875
CSO: 1863/196

UDC 681.51:007.5

INTELLIGENT DATA ACCESS SYSTEMS (REVIEW)

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 (signed to press 6 May 82, after revision 21 Sep 82) pp 66-72

[Article by Vladimir Ivanovich Polyakov, "Intelligent Data Access Systems (Review)", in the section "Data Banks and Information Retrieval Systems"]

[Excerpts] Intelligent data access systems are intended to provide interaction of a nonprogrammer user with databases. These systems are a subclass of conversational systems [1], also called question-response [2] and intelligent data bank [3] systems. The input language in such systems is generally a natural language, limited by lexical composition to a certain (frequently narrow) subject area. However, this attribute is not the distinguishing feature of intelligent data access systems (languages very far removed from a natural one can be used, but not requiring special programming knowledge from the user). The characteristic feature of such systems is a base of knowledge containing a description of the structure and logic of the subject area in which the system operates.

QUESTION-ANSWER [21] is a system with logical representation of knowledge about the subject area. In contrast to QA3, the system axioms and questions are specified not in the predicate calculation language, but in that of the list of properties. The elementary construction of this language is the triplet: <characteristic> <object name> <value> System axioms are divided into two types: specific facts, and derivation rules. For convenient communication with the system, all the input information is specified using lexemes of the natural language. The answers found in the search process are added to the system's knowledge to provide maximum completeness of the answer.

The functions of DILOS (dialog logic system) [22,23] cover the possibilities of intelligent data access systems. In its characteristics, we shall deal only with the components providing access to data. Knowledge in DILOS is represented as a semantic network, specified in the F-language [24], also used as the internal form of representing requests for information search. The input statement is processed in several stages, each realized by one or more processors. The linguistic processor converts phrases from the natural language to the F-language. Then the semantic processor operates, whose task is semantic interpretation of the F-expressions obtained at the preceding stage [7]. Semantic interpretation narrows the range of search for the required information, using the description of "focus" of dialog as a system of frames. The semantic processor starts the information processor

(to search for information in databases) or logic processor (if the information needed is not in an explicit form in the bases) as needed.

The architecture of POET (economic text processing program) [25,26] is almost the same as that of the TORUS system, but the operating algorithms of individual components have several differences (often major). For instance, syntactic analysis is done according to the conventional scheme of text analysis in a natural language [27] according to the algorithm developed [28]. As in the TORUS system, the result of the linguistic processor's work is a semantic network, which is then displayed to the knowledge network at the semantic interpretation stage. The last stage is search in the databases. The response is output in a natural language, and the semantic network of the request used substantially in its formation.

MIMIR [29,30] is a dispatcher system. Its distinguishing features are: first, so-called active semantic networks (M-networks) are used to represent knowledge [31]; second, when new information is entered it is not only monitored using the accumulated experience (deductive derivation), but the system's experience is formed based on inductive derivation procedures; and third, the semantic network is divided into parts (problem spheres). During the answer to the request, a network is assembled from the problem spheres needed for the answer. The requests are formed in the input language YaDRO, whose lexicon is limited to natural language lexemes in a standard form, while the syntax is fixed in the sense that the question begins with an interrogatory word (WHAT, WHICH), then the rema of the request must be defined (what is asked), and finally, the subject (condition of the question). Lexical processingof the request consists of displaying it on a semantic network (rema and subject are associated with certain network nodes). During semantic processing, nodes corresponding to the subject are excited. If the nodes corresponding to the rema are excited during propagation of the excitation over the network, then they are used to form the answer, which is generally a list of objects.

DISPUT (dialog information reference system for planning and management in transport) [32] not only provides access to data, but is also used for planning (optimization models can be used in its operation). As with the DILOS description, we shall consider only that part of the system related to data access. This part includes two processors: linguistic and pragmatic. The linguistic processor translates the request from the limited natural language to the internal representation according to the grammar developed for the specific subject area and realized by the procedure. Linguistic processing of the request uses a semantic network describing the subject area. The pragmatic processor translates the request from the internal form to the actual parameters of the database scheduler, and ensures independence of the linguistic processor from the database with which it will work.

ZAPSIB (request for reference information base) [33] is a family of systems, providing user communication with databases in a natural language and distinguished by their ability to process more complex requests. According to the authors, this design's task is to develop a basic set of program compatible functional modules, enabling a system to be assembled directly for a certain customer. The special program system STEND has been developed to adapt the system to customer needs [34]. Any system from this family consists of a linguistic processor, translating

the request from the natural language to an internal form, and a program module, converting this request to a reference to a relational SUBD [database management system]. The input statement is translated to the internal form by a dictionary, in which each word is connected with semantic information, and a set of rules of the form NAME: CONDITION OF APPLICABILITY→OPERATOR. The system can be adjusted to a specific subject area by complicating the set of rules and using various analysis methods (single- and multivariant).

The Koz'min system [35] is interesting in that the semantics of the database elements are described from the procedural standpoint; i.e., by specifying a set of allowable procedures on them, which are stored in the database. More precisely, in the database are stored prototype frames, containing semantic-pragmatic information on a given class of objects, while the objects in the databases are considered as examples of these frames. The slots of the prototype frame are divided into two groups:

> slots whose names are the objects' properties, while the values of these properties are stored directly in the databases; and
>
> slots whose values are procedures typical of the given class of objects; the properties corresponding to them are not stored in the databases, but computed by these procedures.

A Typical Structure. It is clear from the examples given that the structures of intelligent data access systems can differ from each other quite markedly. Nonetheless, a generalized structure can be suggested such that removing several elements from it can yield the structural scheme of any system. The typical structure we propose is represented in a diagram, and is the result of generalizing the structures of specific systems, some of which have been examined above. In its synthesis, we also considered similar systems proposed by other authors [1,3]. We shall now give a brief description of the individual elements of data access systems.

As already noted, a base of knowledge is a necessary element of any system, but it is a distributed element. The system's knowledge of the given subject area is defined by a dictionary (more accurately, by that part of it describing the semantics of the dictionary units) and the description of the subject area, for which various formalisms can be used: procedural (BASEBALL, SHRDLU), logic (QA3, LUNAR, QUESTION-ANSWER), and semantic (TORUS, DILOS, POET, MIMIR, Koz'min system). Some systems also use a grammar closely related to the given subject area (DISPUT), and employ the database as a component of the base of knowledge (ROBOT).

Linguistic Processor. This processor is found in almost all the systems (save only the QA3 of those considered). Its task is to produce a semantic representation of the input message. It also defines the type of request for data search, specifying for the answer driver the procedure for processing the data displayed by the search processor. The input message is usually formulated as a subset of the natural language, though a formal language (QUESTION-ANSWER) can also be used. The structure and classification of linguistic processors are described thoroughly in a detailed review [36]. We shall simply point out that the systems considered use the following

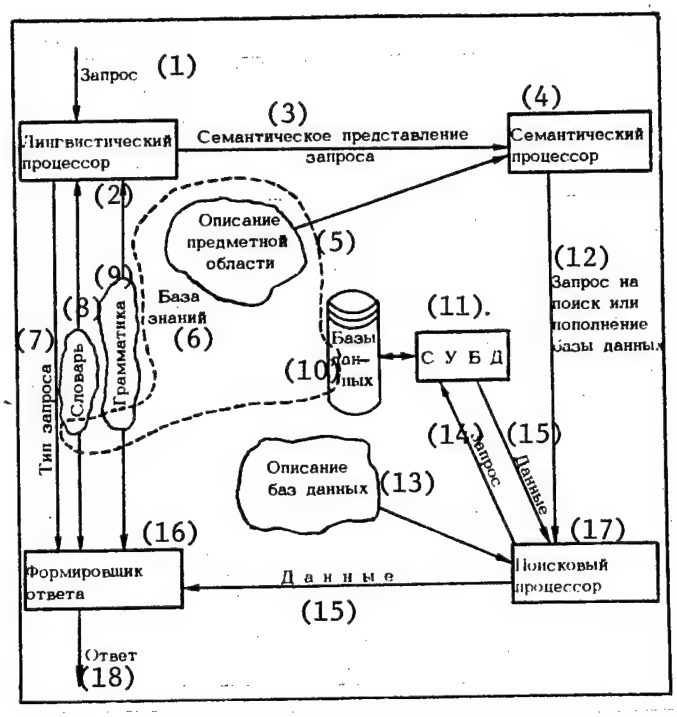


Fig. 1. Typical Structure of an Intelligent Data Access System

Key:

1. Request
2. Linguistic processor
3. Semantic representation of request
4. Semantic processor
5. Subject area description
6. Base of knowledge
7. Type of request
8. Dictionary
9. Grammar
10. Databases
11. Database management system
12. Search request or database update
13. Database description
14. Request
15. Data
16. Answer driver
17. Search processor
18. Answer

types of linguistic processors: syntactic-semantic (DISPUT, BASEBALL, SHRDLU, REQUEST, LUNAR, ROBOT), semantic-syntactic (ZAPSIB, RENDEZVOUS, DILOS, TORUS, MIMIR), and balanced (POET).

Semantic Processor. The main task of this processor is to interpret the semantic representation of the input message. The interpretation process itself depends primarily on whether the input message is a request for data search or a request to modify the description of the subject area or databases. If the semantic representation of a request for data search comes to the semantic processor's input, then the substructures coinciding (in a certain sense) with this request are sought in the description of the subject area. In systems lacking databases (QA3, SHRDLU, QUESTION-ANSWER, MIMIR), with all the facts stored in the subject area description, these substructures contain the answer to the request; in other systems, they contain information on which databases should be processed to obtain the answer. In describing the semantic processors, we shall assume that a request for data search comes to the system input, since many of the systems described can operate only in this mode.

The interpretation process depends largely on which formalism is used to describe the subject area and specify the semantic representation of the input message. Semantic processors of data access systems can thus be divided into three classes according to the formalism used: logic, procedural, and semantic proper.

The logic formalism is used in the QA3, LUNAR, and QUESTION-ANSWER systems. In them, the semantic representation of the input message is a theorem in the formal system used to describe the subject area. The QA3 uses a first order calculation of the predicates; LUNAR, a similar language, MRL; the QUESTION-ANSWER system uses a language of property lists. Interpretation consists of proving the given theorem, using either the Robinson resolvency principle (QA3) or similar methods for the appropriate calculations (LUNAR, QUESTION-ANSWER).

The procedural formalism is used in the SHRDLU system, in which the subject area description is specified in the PLANNER language, while the question is converted to a theory in the same language. The proof of the theorem is conducted using derivation mechanisms built into this language's interpretor.

The semantic formalism is used in the TORUS, POET, MIMIR, DILOS and Kuz'min systems. However, within it are used such very different methods of subject area description as the semantic network (TORUS, POET), M-network (MIMIR), and frames (DILOS, Koz'min system).

In systems using a semantic network to describe the subject area, the representation of the input message is also a semantic network. In data search requests, some of this network's vertices can be marked by the sign "?". Operation of the semantic processor in such systems consists of finding the mapping of the request network to the semantic network describing the subject area. When the required mapping cannot be constructed, deductive procedures converting the semantic representation of the request or adding implicit facts to the semantic network are used. In the TORUS system, for example, it is assumed that the properties of an object are inherited by all the objects below the given one, in the hierarchy of relations "subset exists" and "element exists". It should be point out, however, that such models do not have formal models of the deductive logical derivation, similar to the resolvency principle for systems with a logic formalism. Formation of the output structure generally uses only those vertices of the semantic network in which the request vertices marked by "?" are mapped.

In the MIMIR system, using the M-network to describe the subject area, the semantic processor's task is to excite the network nodes corresponding to the request subject, and propagate the excitation over the network. The response includes all the excited vertices corresponding to the request rema.

In the Koz'min system and DILOS, semantic representation of the request is a copy of the frame for which the semantic processor searches in the subject area description for a prototype frame, which can be synthesized of existing ones using information on the connection of frames by the relations "cause exists", "part exists", and others. After the prototype frame is found (constructed), it is known which elements of the databases must be chosen for the answer to the question, or which procedures must be performed to obtain the answer.

(1) Виды процессоров	Наименование систем (2)													
	BASEBALL	QA3	SHRDLU	RENDEZ-VOUS	LUNAR	TORUS	REQUEST	ROBOT	ВОПРОС-ОТВЕТ	ДИЛОС	ПОЭТ	МИМИР	ДИСПУТ	ЗАПСИБ
(9) *Лингвистические процессоры:*									(3)	(4)	(5)	(6)	(7)	(8)
синтактико-семантический	+		+		+		+	+					+	
семантико-синтаксический				+		+				+		+		+
сбалансированный											+			
Семантические процессоры:														
логический (10)		+			+				+					
процедурный			+											
семантический						+				+	+	+		
Поисковые процессоры: (11)														
специализированный	+				+						+			
СУБД-ориентированный				+		+		+					+	
настраиваемый														+
Формировщик ответов: (12)														
на естественном языке			+			+					+			
в виде таблиц или списков свойств	+	+		+	+		+	+	+	+		+	+	+

Fig. 2. Classification of the Components of Intelligent Data Access Systems

Key:

1. Processor types
2. System name
3. QUESTION-ANSWER
4. DILOS
5. POET
6. MIMIR
7. DISPUT
8. ZAPSIB
9. Linguistic processors: syntactic-semantic, semantic-syntactic, balanced
10. Semantic processors: logic, procedural, semantic
11. Search processors: dedicated, SUBD-oriented, adjustable
12. Answer former: in natural language, as tables or lists of properties

Search Processor. In some data access systems, this processor is missing completely. In the QA3, QUESTION-ANSWER and MIMIR systems, all the information is stored in the subject area description, and there are no databases. In other systems (REQUEST, DILOS), using LISP, the data are indistinguishable from the programs and the search consists of interpreting the expression obtained at the output of the linguistic processor (REQUEST), or executing a series of routines started by the semantic processor (DILOS). In many systems, however, the search component can be separated as an independent structural unit, including the search processor, database description, SUBD and the data themselves. Search processors can be conditionally divided into the following classes: dedicated, SUBD-oriented, and adjustable.

Dedicated search processors are used in the BASEBALL, LUNAR and POET systems. They are characterized by an absence in the structure of the description of

the databases and SUBD's. These systems' search processors are adjusted to a specific data structure, and must be changed when moving to another subject area.

SUBD-oriented search processors are characterized by the fact that they generate a request either to a specific SUBD (TORUS to ZETA, ROBOT to ADABAS, DISPUT to a specially developed SUBD), or in the language used by the SUBD class (for example, in RENDEZVOUS requests are generated in the alpha-language).

Adjustable search processors are characterized by the capability of generating requests to various SUBD's of a certain class (or even to SUBD's of different classes), using the database description. With some stretching, the search processor of the ZAPSIB system can be included in this class, in which it is generated in the process of adapting the specific system to the customer's needs.

Formation of Answers. During operation, the data access system can communicate with the user to eliminate ambiguities arising while processing the request. Standard phraseology is generally used for this, into which words or combinations unfamiliar to the system are inserted. All the systems can be divided into the following two classes by method of forming the results of a search in the databases:

systems forming an answer in a natural language (SHRDLU, TORUS, POET); in the TORUS and POET systems, common answers are used, but the procedure for generating the answer text is considerably simplified by primarily using a semantic representation of the input message when generating the answer; and

systems producing an answer as a list of properties or tables. Obtaining the answer in this form generally yields no objections even from nonprogrammer users, and considerably simplifies the structure of the answer former. Representing the results of a search as a text in a natural language is also obviously unacceptable where the volume of data displayed is very large.

Conclusion: In conclusion, it would seem useful to reduce the proposed classification of specific intelligent data access systems to a compact form. This is done in the table above, where the columns correspond to the systems, and the lines are combined into groups corresponding to the individual system processors. The "+" sign means that a processor belonging to the corresponding group is used in the given system.

BIBLIOGRAPHY

1. Pospelov D.A., "Dialog Systems: Difficulties and Successes", in "Semantika i predstavleniye znaniy. Trudy po iskusstvennomy intellektu" [Semantics and Representation of Knowledge. Works on Artificial Intelligence], Tartu, Gos. un-t, 1980, Part 2, pp 82-98.

2. Grin, K., "Proof of Theorems Using the Resolution Rule as the Basis for Building a Question-Answer System", in "Iskusstvennyy intellekt" [Artificial Intelligence], M, Mir, 1973, pp 275-309.

3. Kuzin L.T., "Intelligent IBD Data Banks", VOPROSY KIBERNETIKI. INTELLEKTUAL'NYYE BANKI DANNYKH, 1979, Vol 55, pp 5-16.

4. "BASEBALL: A Program That Automatically Answers Questions", B. Grin (Jr.), A. Vulf, K. Khomskiy, K. Loferi, in "Vychislitel'nyye mashiny i myshleniye" [Computers and Thought], M, Mir, 1967, pp 221-223.

5. Khant E., "Iskusstvennyy intellekt" [Artificial Intelligence], M, Mir, 1978, 558 pages.

6. Pertsova N.N., "Methods of Representing Subject Areas During Automatic Text Processing (Review)", IZVESTIYA AN SSSR. TEKHNICHESKAYA KIBERNETIKA, 1977, No 5, pp 51-59.

7. Bryabrin V.M., "Structural Descriptions as the Basis for Semantic Interpretation of Natural Language Texts", in "Vzaimodeystviye s EVM na yestestvennom yazyke" [Interaction With the Computer in a Natural Language], Novosibirsk, VTs SO AN SSSR, 1978, pp 166-181.

8. Vol'fengagen V.A., Vyaz'min S.A., Rybina G.V., "Frames in a Knowledge Representation System", in "Informatsionno-programmnoye obespecheniye sistem iskusstvennogo intellekta" [Information and Program Support of Artificial Intelligence Systems], M, MDNTP, 1978, pp 18-26.

9. Minskiy M., "Freymy dlya predstavleniya znaniy" [Frames for Representing Knowledge], M, Energiya, 1979, 152 pages.

10. Vinograd T., "Programma, ponimayushchaya yestestvennyy yazyk" [Programs Understanding a Natural Language], M, Mir, 1976, 294 pages.

11. Pertsova N.N., "One Direction in Research in the Field of Artificial Intelligence ("Robot" of T. Vinograd)", NAUCHNO-TEKHNICHESKAYA INFORMATSIYA. SER. 2. INFORMATSIONNYYE PROTSESSY I SISTEMY, 1975, No 8, pp 30-35.

12. Codd E.F., "Seven Steps to Rendezvous With the Casual User", Data Base Management, Amsterdam et al, North-Holland Publ. Comp., 1976, pp 179-200.

13. Woods W.A., "Semantics and Quantification in Natural Language Question Answering", Advances in Computers, 1978, Vol 17, pp 1-87.

14. Woods W.A., "Network Grammars for Analyzing Natural Languages", KIBERNETICHESKIY SBORNIK. NOVAYA SERIYA, 1976, Vol 13, pp 120-158.

15. "TORUS: A Data Management System Understanding Natural Language", "Trudy 4-y Mezhdunarodnoy ob'yedinennoy konferentsii po iskusstvennomy intellektu. T. 6. Obshcheniye s EVM na yestestvennom yazyke" [Proceedings, Fourth International Joint Conference on Artificial Intelligence. Vol. 6. Communication With the Computer in a Natural Language], M, 1975, pp 6,116-6,137.

16. Bruce B., "Case System for Natural Language", Artificial Intelligence, 1975, Vol 6, No 4, pp 327-360.

17. Koen F., Milopulos D., Bordzhida A., "Some Aspects of Representation of Knowledge", in "Trudy", op. cit., Vol. 2, "Metody predstavleniya zadach, metody poiska resheniy, evristicheskiye metody" [Methods of Task Representation, Methods of Solution Search, Heuristic Methods], M, 1975, pp 2.42-2.62.

18. Petric S.R., "On Natural Language Based Computer Systems", IBM J. Res. Develop., 1976, Vol 20, No 4, pp 314-325.

19. Plath W.J., "REQUEST: A Natural Language Question-Answering System", Ibid., pp 326-335.

20. Harris L.R., "Using the Data Base as a Semantic Component to Aid in the Parsing of Natural Language Data Base Queries", J. Cybernt., 1980, Vol 10, No 3, pp 77-96.

21. Bukhshtab Yu.A., Kamynin S.S., "Organization of a Deductive System Answering Questions", PROGRAMMIROVANIYE, 1975, No 6, pp 83-90.

22. Bryabrin V.M., "Dialogovaya informatsionno-logicheskaya sistema" [An Interactive Information-Logic System], M, Nauchn. sovet po kibernetika, 1977, 24 pages.

23. Bryabrin V.M., Senin G.V., "Rukovodstvo k sisteme DILOS--BESM-6" [Guide to the DILOS--BESM-6 System], M, Nauchn. sovet po kibernetike, 1977, 46 pages.

24. Bryabrin V.M., "The F-Language: A Formalism for Representing Knowledge in an Intelligent Interactive System", PRIKLADNAYA INFORMATIKA, 1981, Vol 1, pp 73-103.

25. Baklanov V.M., Popov E.V., ""Understanding" Phrases of Limited Russian", IZVESTIYA AN SSSR. TEKHNICHESKAYA KIBERNETIKA, 1978, No 4, pp 66-74.

26. Popov E.V., "A System of Interaction With the Computer in Limited Russian", PROGRAMMIROVANIYE, 1978, No 4, pp 66-76.

27. Mel'chuk I.A., "Opyt teorii lingvisticheskikh modeley "smysl-tekst"" [The Experience of the "Meaning-Text" Theory of Linguistic Models], M, Nauka, 1974, 314 pages.

28. Popov E.V., "An Effective Algorithm of Syntactic Analysis of Russian Phrases", IZVESTIYA AN SSSR. TEKHNICHESKAYA KIBERNETIKA, 1978, No 5, pp 84-92.

29. Lyubarskiy Yu.A., "Representation of Knowledge About a Control Object in Dispatcher Information Systems", PROGRAMMIROVANIYE, 1978, No 1, pp 41-50.

30. Lyubarskiy Yu.A., "A Dispatcher Question-Answer System With a Virtual Semantic Network", PROBLEMY BIONIKI, 1980, Vol 24, pp 86-93.

31. Amosov N.M., Kasatkin A.M., Kasatkina L.M., Talayev S.A., "Avtomaty i razumnoye povedeniye. Opyt modelirovaniya" [Automata and Rational Behavior. The Experience of Simulation], Kiev, Naukova dumka, 1973, 374 pages.

32. "An Interactive Information Reference System for Planning and Management in Transport Systems DISPUT. Part 1. Construction Principles and Description of the Operation of the DISPUT System. Part 2. Realization of the DISPUT System", A.S. Belen'kiy, L.I. Mikulich, N.Ya. Naydenova, A.Ya. Chervonenkis, AVTOMATIKA I TELEMEKHANIKA, 1981, No 3, pp 152-162; No 5, pp 169-180.

33. Narin'yani, A.S., ""Lingvisticheskiye protsessory ZAPSIB. Chast' 1. Zadacha proyekta. Chast' 2. Obshchaya skhema i osnovnyye moduli" [The ZAPSIB Linguistic Processors. Part 1. Project Task. Part 2. General Scheme and Basic Modules], Novosibirsk, 1979, 22 pages; 48 pages (VTs SO AN SSSR, preprints Nos. 199 and 202).

34. Levin D.Ya., "STEND: sistema adaptatsii lingvisticheskikh protsessorov" [STEND: An Adaptation System of Linguistic Processors], Novosibirsk, 1980, 30 pages (VTs SO AN SSSR, preprint No 238).

35. Koz'min A.A., "Predstavleniye znaniy v sisteme intellektual'nogo dostupa k dannym na osnove pragmatizatsii semanticheskikh protsedur" [Representation of Knowledge in Intelligent Data Access Systems Based on Pragmatization of Semantic Procedures], M, Nauchn. sovet po kibernetike, 1980, 6 pages (Materials of the International Conf. on Art. Intelligence, Repino, 1980).

36. Khoroshevskiy V.F., "Intelligent Interactive Systems", IZMERENIYA. KONTROL'. AVTOMATIZATSIYA, 1980, No 5-6, pp 45-55.

9875
CSO: 1863/196

UDC 681.3.06

GRAPHICS PACKAGE GRAS: BUFFERED OUTPUT

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 (signed to press 24 Nov 82) pp 96-101

[Article by Yuriy Vladimirovich Kol'tsov, Vladimir Vyacheslavovich Manako and Andrey Ivanovich Nikitin, "The Graphics Package GRAS. Buffered Output", in the section "Application Program Packages"]

[Excerpts] Introduction. Standardization of graphic software is a main condition for enhancing the efficiency of using machine graphics tools in various fields. A major step in this direction involved the appearance of the graphic standard designs Core System (USA) and GKS (FRG). However, subsequent work on graphic systems here and abroad considering the assumptions and recommendations of these projects indicates that work to standardize machine graphics software, especially to standardize various graphic output components, is still far from completed.

This article describes certain issues of design, functional capabilities and implementation of the buffered output (BUV) of the GRAS package [1,2], designed with consideration for the assumptions and recommendations of the GSPC ACM/SIGGRAPH graphic standards projects [3,4]. These projects are incomplete and contradictory in many ways, which raises the need to strictly observe basic principles and criteria in developing a common architecture of the package and its individual levels and functions. In developing the GRAS BUV, we thus relied on the criteria and principles formulated in [1], refined, supplemented and specified in greater detail where needed.

BIBLIOGRAPHY

1. Manako V.V., "The GRAS Standard Graphic Packet. Design. Derivation Basis", USiM, 1981, No 3, pp 48-54.

2. Kol'tsov Yu.V., Manakov V.V., "Buffered Output in the Standard Graphics Package GRAS", "Prikladnoye programmirovaniye" [Application Programming], Kiev, IK AN USSR, 1982, pp 79-83.

3. "ACM/SIGGRAPH: Status Report of the Graphics Standards Planning Committee of ACM/SIGGRAPH", Computer Graphics, 1977, Vol 11, No 3, pp 1-130.

4. "ACM/SIGGRAPH: Second Status Report of the Graphics Standards Planning Committee of the ACM/SIGGRAPH", Computer Graphics, 1979, Vol 13, No 3, pp 1-276.

5. Knut D., "Iskusstvo programmirovaniya dlya EVM. Sortirovka i poisk" [The Art of Computer Programming. Sort and Search], M, Mir, 1978, Vol 3, 842 pages.

9875
CSO: 1863/196

UDC 681.3.06

GRAPHICS PACKAGE FOR DATA ANALYSIS. IMAGE FORMATION TOOLS

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 (signed to press 18 Nov 81, after revision 23 Aug 82) pp 104-109

[Article by Stanislav Vladimirovich Klimenko, Valentin Nikolayevich Kochin and Aleksandr Vadimovich Samarin, "A Graphics Package for Data Analysis. Image Formation Tools", in the section "Application Program Packages"]

[Excerpts] Introduction. The graphics package ATOM is a problem-oriented package of subroutines, basically designed for tasks of analyzing experimental physics data. It is based on the hierarchical principle [1,2]. The lower level consists of graphic device drivers, the next level is the hardware-independent general purpose graphics package, and the upper levels are problem-oriented. Each software level corresponds to a certain degree of abstraction of data representation, and is a virtual device for the operation of a higher level. This approach provides terminal independence of application programs, and adaptability of the package itself relative to the class of tasks handled and hardware configuration.

This work considers image formation tools in the general purpose graphics package. The basic goal is to study the features of the image construction process in data analysis problems, and choose the optimal set of basic procedures for the given class of problems.

The set of functions performed by the general purpose subroutines must allow the most convenient method of constructing any images encountered while using machine graphics for data analysis problems. In choosing the set of required functions, we started with the recommendations of the working group ACM/SIGGRAPH GSPC, proposing the Core Graphics System [3,4] set of basic graphics procedures as the standard. These proposals were reworked and supplemented considering the features of using machine graphics in data analysis problems [1,2,5]. Naturally, in each specific area of application of machine graphics the basic set includes higher level functions than in the Core System. This approach ensures the most convenient set of functions, enabling effective realization.

The basic features of the ATOM package's image formation tools are dictated by two contexts in constructing images in data analysis problems, corresponding to data representation and formation of drawings [5]. Each context has its own system of coordinates and its own (generally different) set of primitives. In the context

of data representation, the most usable are primitives constructing various lines and markers. In the drawing formation context, the essential one is construction of various texts.

Conclusion. The image formation tools considered in this article belong to general purpose subroutines within the ATOM problem-oriented package, designed for use in problems of analyzing experimental physics data. The subroutines are realized in FORTRAN and used at the IFVE [Institute of High Energy Physics] on the DEC-10 and ICL-1906. The programs were generated in close cooperation with the users, enabling the package to be checked for working ability, convenience and efficiency when applied to specific tasks.

The basic features of the subroutines described are dictated by two contexts in image construction, corresponding to data representation and drawing formation. To work in these contexts, convenient coordinate systems were introduced, the set of coordinate and coordinate-free graphic primitives and their corresponding attributes defined, and the capabilities of the basic graphics system expanded to work with texts. The mechanism of indirect addressing is used when calling up subroutines to ensure broad functional capabilities of the graphics package.

BIBLIOGRAPHY

1. "Software of Machine Graphics for a Physics Experiment", L.G. Kaminskiy, S.V. Klimenko, V.N. Kochin, A.V. Samarin, "Dialogovyye vychislitel'nyye kompleksy: Materialy 2-go vsesoyuz. soveshchaniya" [Interactive Computer Complexes: Materials, 2nd All-Union Conference], Serpukhov, 1979, pp 288-293.

2. "The Graphics Package ATOM. Structure and Basic Principles", L.G. Kaminskiy, S.V. Klimenko, V.N. Kochin et al, Serpukhov, 1981, 25 pages (IFVE, preprint 81-156).

3. "ACM/SIGGRAPH: Status Report of the Graphic Standards Planning Committee of ACM/SIGGRAPH", Computer Graphics, 1977, Vol 11, No 3, pp 1-130.

4. "ACM/SIGGRAPH: Second Status Report of the Graphic Standards Planning Committee of ACM/SIGGRAPH", Computer Graphics, 1979, Vol 13, No 1, pp 1-276.

5. Cahn D.U., Johnston N.E., Johnston W.E., "A Response to the 1977 GSPC Core Graphics System", Computer Graphics, 1979, Vol 13, No 1, pp 57-62.

6. N'yumen U., Sprull R., "Osnovy interaktivnoy mashinnoy grafiki" [Basic Interactive Machine Graphics], M, Mir, 1976, 574 pages.

7. "PLOT-10/TCS: Standard FORTRAN Subroutine Package", Beaverton, TEXTRONIX Inc., 1976, 120 pages (No 062-1474-01).

8. Miller R., "GD-3, CERN Computer Program Library Long Write-Up", Geneva, CERN, 1976, 128 pages.

9. Klimenko S.V., Kochin V.N., "One Method of Representing Surfaces in Machine Graphics", PROGRAMMIROVANIYE, 1981, No 2, pp 52-62.

9875
CSO: 1863/196

UDC 681.322.06

ORAKUL-1 INFORMATION REFERENCE SYSTEM

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 (signed to press 1 Feb 82, after revision 18 Nov 82) pp 84-88

[Article by Vladimir Avdeyevich Kushnirov, Frank Andreyevich Levchenko and Vladimir Iosovich Sarkisyan, "The ORAKUL-1 Information Reference System", in the section "Data Banks and Information Search Systems"]

[Excerpts] Introduction. The use of a special-purpose information reference system (ISS) to organize the computation process at a VTs [computer center] can be viewed as one solution to the problem of enhancing the efficiency of computer use.

It is known that after abnormal termination of a program on a Unified System [YeS] computer the programmer obtains a code of the OS YeS [YeS operating system] diagnostic message: the termination code, message of the link editor or job control language, translator, etc. In addition, at the computer console or in the error log the computer records the system codes of malfunctions of the processor, peripherals and communication channels, and the machine error record codes for the VTs maintenance personnel.

Choosing the correct algorithm for further processing requires a knowledge of the text portion of the diagnostic message codes, system codes, service routines and instructions on organizing the computation process. We shall look at two of the main problems encountered by the programmer and service personnel at a VTs.

First, the total number of diagnostic message codes and system codes approaches four thousand; the text portion of each of them has an average of some 16 lines of printed text. It is impossible to store this amount of text; moreover, a search takes not only time, but also a considerable amount of documentation. For instance, the OS YeS version 2.1 contains 55 volumes of documentation; the 4.1 version, 83; and the 6.1 version, 120.

Second, to save computer resources when performing work at a given VTs, a large number of service routines and instructions on the computation process must be known. For instance, the VTs IK AN USSR [computer center of the Institute of Cybernetics of the UkSSR Academy of Sciences] presently uses over 300 procedures related to computation management, data manipulation, etc.

Obviously, only a system capable of providing programmers and service personnel with the necessary information can serve as a tool for improving the organization of the computation process at a VTs.

Since such an ISS is intended to realize service functions in parallel with a continuous computation process, the computer resources and reaction time to a user enquiry must be minimized.

Existing SUBD's [database management systems] do not fully meet the requirements imposed on such an ISS. Such SUBD's as OKA, PAL'MA, BANK and TVER' require substantial main memory resources (at least 256K bytes), due to their universality. In addition, these systems' reaction time to an enquiry is considerable, while BANK and TVER' systems lack remote access resources.

The need has thus arisen to create an interactive ISS to execute service functions when managing the computation process on a YeS computer.

This article describes the interactive ISS ORAKUL-1, developed at the SKTB PO IK AN USSR [special design technology office of the production association of the IK AN USSR].

Structure and Functional Diagram of the ORAKUL-1 System. An analysis of existing ISS's [1-13] shows that they can be divided into data retrieval and document systems by construction principle and method of outputting responses to user enquiries.

Data retrieval systems output specific answers, containing only actually existing data, to specific data enquiries. Document systems not only provide statistical data to the user, but also perform a certain processing of the data in a given subject area.

The ORAKUL-1 system is a data retrieval ISS, operating in the OS YeS medium.

Realization of the ORAKUL-1 System. As noted above, the ISS ORAKUL-1 was developed and realized at the SKTB PO IK AN USSR. The system is controlled by the OS YeS, versions 4.1, 6.1 and 6.1 01, and requires a working memory of 102K bytes and external memory of 1 to 8 magnetic disks (7.25, 29 and 100M bytes). One to 16 display terminals and one printer can be connected to it.

The system's database presently holds around 2,000 OS YeS messages. The system's response time to a user enquiry on the YeS1060 is 1 to 1.8 seconds (even with concurrent operation with a time-sharing system and with batch processing). The system uses YeS7906 and YeS7920 terminals. Work to connect user stations to the system is continuing.

To obtain the information needed, the system user employs a keyboard display to type in the message code on the screen and sends it to the system. Two versions of system reaction are possible: a message with the indicated code is absent; the code and text of the message are displayed on the screen.

In the first case, the user moves on to the next enquiry; in the second, all or part of the message is examined in the "scroll text" mode, while the information is on the display screen. Transfer from automatic data motion control to manual can be done at any time.

Review of the message called up and transfer to another enquiry's data can be achieved easily by specifying an instruction without parameters.

An unlimited number of hard copies of the data called up can be obtained at any time on the printer (alphanumeric or part of the display).

Reference information can also be obtained in another manner. The user can choose and note the names (codes) of messages interesting him and send them to the VTs. The computer operator, having specified the indicated names (codes), will obtain a listout containing the information needed, and then send it to the user.

The ORAKUL-1 system can also be used to output annotations to books in the library holdings, obtain information on the plans of organizations, ministries and departments, and output annotations to patents.

Let us look at a more complicated organization of data in the base. Over 600 patent annotations from various countries for 1980 were entered in the system (about 3,500 patent annotations can be written onto one 29M byte magnetic disk).

Having keyed in the PATENT code, the user obtains on the display screen a list of patent topic sections with codes corresponding to each one: PIBM, computers; PUVV, input-output devices; PRAS, algorithms and devices for recognizing patterns, characters and monitor resources; PUST, control devices and memory devices; etc.

Having chosen the section interesting him, the user specifies its code on the screen. After executing this enquiry, the screen displays the titles of the patents in that section, with the corresponding codes. For example:

> PU1731 Monitoring system of data processing device. Kan. SETsUO et al (Fudzitsu k.k.). Jap. pat., cl. 97 (S02, G11 C 9/06), No 53-21622.
> PU2042 Method and device for recognizing and processing alphanumeric characters. Application FRG, cl. G06 K 9/13, G06 F 3/00, No 2717596.
> PU2234 Equipment for protection from mutual effect of processors in a multiprogram system. Pat. USA, cl. 364/200, (G06 F 15/16), No 4096561.

The user of the ORAKUL-1 system can call up the annotation to any patent(s) on the screen by specifying the appropriate code (identifier,containing six symbols). If the text of the annotation called up does not fit on the screen, it can be reviewed in the "scroll text" mode.

A user can be trained to use the ORAKUL-1 system in several hours.

The ORAKUL-1 system has been introduced at several organizations of the Ukrainian SSR. The economic effect from the introduction is 427,500 rubles a year.

BIBLIOGRAPHY

1. Glushkov V.M., Stogniy A.A., Afanas'yev V.N., "Avtomatizirovannyye informatsionnyye sistemy" [Automated Information Systems], M, Znaniye, 1973, 286 pages.

2. Stogniy A.A., Zaytsev, N.G., "Automated Information Reference Systems: Their Functions, Characteristics and Basic Requirements", KIBERNETIKA, 1969, No 4, pp 12-15.

3. "PAL'MA: A Database Management System With a Complex Architecture", F.I. Andon, A.A. Bakayev, R.P. Kramarenko, T.V. Kostruba, "Banki dannykh: Materialy vsesoyuz. konf." [Data Banks. Materials, All-Union Conference], Tbilisi, 1980, pp 24-25.

4. Dzhoyner R., Tanstoi K., "EVM dopolnyayet protsess prinyatiya resheniy" [The Computer Supplements the Decision Making Process], M, Mir, 1973, 220 pages.

5. Devis R.M., "Metody razrabotki voyennykh informatsionnykh sistem" [Methods of Developing Military Information Systems], M, Mir, 1966, 196 pages.

6. Lankaster F.U., "Informatsionno-poiskovyye sistemy" [Information Search Systems], M, Mir, 1974, 346 pages.

7. Martin Dzh., "Sistemnyy analiz peredachi dannykh" [System Analysis of Data Transmission], M, Mir, 1970, Vol 2, 238 pages.

8. Solton Dzh., "Dinamicheskiye bibliotechno-informatsionnyye sistemy" [Dynamic Library Information Systems], M, Mir, 1979, 406 pages.

9. "An Automated Information Search System by ADRIS Standard", TEORIYA I PRAKTIKA NAUCHNOY INFORMATSII: EKSPRESS-INFORMATSIYA, VINITI, 1971, No 46, ref. 246, pp 45-55.

10. "An Automated Information Search System of Biology and Medicine of the ATLAS Company (USA)", ibid., 1972, No 12, ref. 7, pp 32-43.

11. "An Information Search System by Frames of the Government Civil Service of Great Britain", ibid., 1972, No 16, ref. 18, pp 14-25.

12. Dworatschek S., "Management: Information Systems', Berlin, New York, Gc Gruter, 1971, 210 pages.

13. Kelly I.F., "Computerized Management Information System", London, Macmillan, 1970, 186 pages.

14. Kleynrok L., "Kommunikatsionnyye seti" [Communication Networks], M, Nauka, 1970, 292 pages.

9875
CSO: 1863/196

RSFSR COMPUTER CENTERS' 10-YEAR PERFORMANCE SUMMARIZED

Moscow VESTNIK STATISKIKI in Russian No 5, May 83 pp 55-59

[Article by Candidate of Economic Sciences N. Yegorov, deputy chief, RSFSR TsSU, and engineer N. Zimenkova and group director, Doctor of Economic Sciences A. Frenkel', Republic Computer Center, RSFSR TsSU: "Analysis of Production Management Activities of the Computer System of the RSFSR TsSU"]

[Text] Steps are now being taken by the computer system of the RSFSR TsSU [Central Statistical Administration] to improve the economic mechanism, which is why the role of indicators describing different aspects of the production management activities of computer centers is growing. In its dynamics, the system of such indicators provides a possibility for evaluating the work of a computer system, revealing some reserves for raising its effectiveness, tracing the basic developmental trends and so on.

Production management activities are influenced primarily by indicators characterizing the dimensions of production: work volume in terms of retail cost ; average listed number of personnel; mean annual cost of fixed productive capital; proportion of the active part of fixed productive capital reflecting, to a certain extent, the equipment availability; the percent of jobs performed by calculator within the total job volume--an indicator characterizing the organizational level of production processes at a computer center; the mean daily computer load, which reflects the effectiveness with which electronic computers are used; the mean daily load of computers characterizing the effectiveness with which tabulators, adding machines, calculators, bookkeeping and invoicing machines are used; pay calculated per man-hour of worked time, interpreted as a material stimulation factor; pay calculated per ruble of the volume of work done--an indicator characterizing the labor-intensiveness of work done at the computer center. Production management activity is also characterized by these indicators: mean annual wages per worker, or labor productivity calculated as the ratio of work volume in terms of retail cost to the average listed strength of all personnel; the output-capital ratio, calculated as the ratio of the work volume in terms of the retail cost to the mean annual cost of fixed productive capital (an indicator characterizing the effectiveness with which the fixed productive capital of the computer center is used); outlays per ruble of the volume of work done, or production cost, calculated as the ratio of production outlays to the annual work volume (the difference between retail cost and production cost represents the computer

center's profit); total profitability, calculated as the ratio of the total balance profit to the mean annual cost of fixed productive capital and standardized working capital, and permitting a comparison between the impact and the utilized reserves.

Let us examine the influence of all these indicators on the work of the computer system of the RSFSR TsSU in 1971-1980 (the dynamics of indicators for the computer system of the RSFSR TsSU in 1971-1980 are shown in Table 1, and the average absolute increments and the mean annual growth rates of these indicators are shown in Table 2).

The mean annual cost of fixed productive capital exhibited the highest growth rate. In the period of analysis, this volume increased by more than sixfold. The mean annual growth rate was 122.7 percent. It should be emphasized that the absolute value of the increment is rather high and stable throughout the entire period. In the 10th Five-Year Plan the mean absolute increment was 27.15 million rubles, in the 10th Five-Year Plan it was 27.5 million rubles, and it was 27.34 million rubles for the entire period under analysis. The growth rate of this indicator was almost three times higher in the 9th Five-Year Plan (35.1 percent) than in the 10th. The decrease in the rate is associated with the fact that in 1971-1975 the computer system of the RSFSR TsSU was intensively equipped with electronic computers; in subsequent years a significant proportion of the computer centers were outfitted with electronic computers, and growth of the number of calculating devices slowed down somewhat; development proceeded mainly due to qualitative reequipment of the system.

The work volume indicator increased at a stable rate, growing by more than four times (Table 1). However, a significant decline occurred in the work volume growth rate during the 10th Five-Year Plan. While in the 9th Five-Year Plan the rate averaged 22.4 percent, in the 10th it was only 11.7 percent. On the whole for 1971-1980, the work volume growth rate was 17.6 percent.

It should be noted that the work volume growth rate was behind the growth rate of the mean annual cost of fixed productive capital, which had an unfavorable effect on the output-capital ratio indicator. Throughout the entire period of analysis its level exceeded that of a previous year on only three occasions (in 1976, 1977 and 1980), and even on these occasions the difference was insignificant. In all other years the output-capital ratio gradually declined, falling by 31.3 percent between 1971 and 1980. While in the 9th Five-Year Plan we observed a sharp decline in the output-capital ratio (the mean annual growth rate was negative, -9.4 percent), in the 10th it remained at approximately the same level. On the whole, meanwhile, the mean annual decrease in the output-capital ratio was 4.1 percent for the period under analysis. All of this can be explained by the fact that expensive equipment introduced for the first time into the computer centers was unable to support the needed increment in work volume at that time. This was the consequence of insufficient utilization of available equipment and slow assimilation of the new equipment.

It should also be noted that the increase in work volume was significantly faster than the growth of the average listed number of personnel as a result

Table 1. Dynamics of the Principal Indicators of the Production Management Activities of the Computer System of the RSFSR TsSU in 1971-1980

(1) Показатели		(2) Годы									
		1971	1972	1973	1974	1975	1976	1977	1978	1979	1980
(3) Объем работ в отпускной стоимости, млн. руб.	а*	46,8	58,1	70,9	87,1	104,9	129,7	150,6	170,1	186,3	201,9
	б	—	124,1	122,0	122,8	120,4	123,6	116,1	112,9	109,5	108,4
(4) Среднесписочная численность персонала, тыс. чел.	а	29,6	36,0	41,9	50,2	58,5	66,4	72,4	74,1	75,3	75,8
	б	—	121,6	116,4	119,8	116,5	113,5	109,0	102,3	101,6	100,7
(5) Среднегодовая стоимость основных производственных фондов, млн. руб.	а	46,6	62,4	84,3	116,6	155,2	187,8	216,5	245,3	272,3	292,7
	б	—	133,9	135,1	138,2	133,2	121,0	115,3	113,3	111,0	107,5
(6) Удельный вес активной части основных производственных фондов, %	а	65,3	71,6	73,4	76,8	80,1	79,6	80,4	80,4	80,3	86,0
(7) Процент работ на ЭВМ в общем объеме работ	а	5,0	5,0	5,6	7,8	9,2	11,9	13,7	15,2	17,1	20,2
(8) Среднесуточная загрузка ЭВМ, ч	а	8,6	9,6	10,0	11,5	6,3	5,6	6,6	6,4	7,1	7,7
	б	—	111,6	104,2	115,0	54,8	88,9	117,9	97,0	110,9	108,5
(9) Среднесуточная загрузка вычислительных машин, ч	а	7,2	7,8	7,1	6,8	6,8	6,5	6,5	6,5	6,6	6,4
	б	—	108,3	91,0	95,8	101,5	94,2	100,0	100,0	101,5	97,0
(10) Заработная плата в расчете на 1 чел.-ч отработанного времени, руб.	а	0,51	0,53	0,55	0,55	0,56	0,56	0,60	0,65	0,68	0,72
	б	—	103,9	103,8	100,0	101,8	100,0	107,1	108,3	104,6	105,9
(11) Расходы зарплаты на 1 руб. объема выполненных работ, руб.	а	0,68	0,69	0,67	0,66	0,65	0,61	0,60	0,59	0,57	0,57
	б	—	101,5	97,1	98,5	98,5	93,8	98,4	98,3	96,6	100,0
(12) Среднегодовая выработка на одного работника, руб.	а	1 584	1 614	1 692	1 737	1 793	1 955	2 081	2 295	2 473	2 665
	б	—	101,9	104,8	102,7	103,2	109,0	106,4	110,3	107,8	107,8
(13) Фондоотдача, руб.	а	1,005	0,932	0,841	0,748	0,676	0,691	0,696	0,694	0,684	0,690
	б	—	92,7	90,2	88,9	90,4	102,2	100,7	99,7	98,6	100,9
(14) Затраты на 1 руб. объема выполненных работ, руб.	а	0,905	0,906	0,892	0,902	0,922	0,908	0,899	0,889	0,875	0,877
	б	—	100,1	98,5	101,1	102,2	98,5	99,0	98,9	98,4	100,2
(15) Рентабельность общая, %	а	8,6	7,8	8,0	6,6	4,0	5,4	6,0	6,8	7,9	8,1

*Here and subsequently: *a*--absolute value; *b*--percent of previous year

Key:

1. Indicator
2. Year
3. Work volume in terms of retail cost, millions of rubles
4. Average listed number of personnel, thousands of persons
5. Mean annual cost of fixed productive capital, millions of rubles
6. Proportion of the active part of fixed productive capital, %
7. Percent of jobs performed by computer with respect to total job volume
8. Mean daily load on electronic computers, hours
9. Mean daily load on calculators, hours
10. Pay calculated per man-hour of worked time, rubles
11. Pay per ruble of work volume, rubles

[Key continued on following page]

12. Mean annual wages per worker, rubles
13. Output-capital ratio, rubles
14. Outlays per ruble of work volume, rubles
15. Total profitability, percent

Table 2. Indicators of the Economic Production Activity of the Computer System of the RSFSR TsSU in 1971-1980

(1) Показатели	1971—1975 гг.		1976—1980 гг.		1971—1980 гг.	
	(2) Средний абсолютный прирост*	(3) Среднегодовые темпы роста, %	Средний абсолютный прирост*	Среднегодовые темпы роста, %	Средний абсолютный прирост*	Среднегодовые темпы роста, %
(4) Объем работ в отпускной стоимости	14,52	122,4	19,40	111,7	17,23	117,6
(5) Среднесписочная численность персонала	7,23	118,6	3,46	103,4	5,13	111,0
(6) Среднегодовая стоимость основных производственных фондов	27,15	135,1	27,50	111,7	27,34	122,7
(7) Удельный вес активной части основных производственных фондов	3,70	105,2	1,18	101,9	2,30	103,1
(8) Процент работ на ЭВМ в общем объеме работ	0,46	116,5	2,20	114,1	1,69	116,8
(9) Среднесуточная загрузка ЭВМ	—0,58	92,5	0,28	108,3	—0,10	97,2
(10) Среднесуточная загрузка вычислительных машин	—0,08	98,5	—0,10	99,6	—0,09	98,7
(11) Заработная плата в расчете на 1 чел.-ч отработанного времени	0,01	102,4	0,03	106,5	0,02	103,9
(12) Расходы заработной платы на 1 руб. объема выполненных работ	—0,01	98,9	—0,016	98,3	—0,012	98,1
(13) Среднегодовая выработка на одного работника	52,25	103,1	174,4	108,1	120,1	106,0
(14) Фондоотдача	—0,08	90,6	0,003	100,0	—0,035	95,9
(15) Затраты на 1 руб. объема выполненных работ	0,004	100,5	—0,009	98,9	—0,003	99,7
(16) Рентабельность общая	—1,15	82,6	0,82	110,7	—0,06	99,3

* The mean absolute increment is given for each indicator in the appropriate units of measurement (see Table 1).

Key:
1. Indicator
2. Mean absolute increment
3. Mean annual growth rate, %
4. Work volume in terms of retail cost
5. Average listed number of personnel
6. Mean annual cost of fixed productive capital
7. Proportion of the active part of fixed productive capital
8. Percent of jobs performed by computer with respect to total job volume
9. Mean daily load on electronic computers
10. Mean daily load on calculators
11. Pay calculated per man-hour of worked time
12. Pay per ruble of work volume
13. Mean annual wages per worker
14. Output capital ratio
15. Outlays per ruble of work volume
16. Total profitability

of stable growth of mean annual wages per worker, which increased by more than a time and a half within the period under examination--from 1,584 to 2,665 rubles at a mean annual growth rate of 6 percent. In the 10th Five-Year Plan the rate of growth of this indicator increased even more--8.1 percent as compared to just 3.1 percent in the 9th. A large difference is also observed in the absolute values of the mean annual increment in wages per worker: 52.3 rubles in the 9th Five-Year Plan and 174.4 rubles in the 10th.

Throughout the entire period of analysis a uniform decrease was observed in the growth rate of the average listed number of personnel in comparison with the previous year. While in 1972 the average listed number of personnel increased by 21.6 percent in comparison with 1971, in 1980 it increased by just 0.7 percent in comparison with 1979. A significant decline in absolute increments in the 10th Five-Year Plan in comparison with the 9th and a correspondingly sharp decrease in growth rate (18.6 percent in the 9th Five-Year Plan and 3.4 percent in the 10th) are typical of this indicator. While work volume, productive capital and the absolute number of employees grew in general, the decrease in the relative increment of the average listed number of personnel also attests to growth in labor productivity and in the work effectiveness of computer centers in the system as a whole owing to intensive factors.

As we know, there are basically two factors that have an influence on growth in work volume--growth in the number of workers (an extensive factor) and growth in labor productivity (an intensive factor). Let us examine how each of these influenced growth in work volume throughout the entire period of analysis (See Table 3) and in individual five-year plans (1971-1975 and 1976-1980). Index analysis was used for the calculations.

Table 3. Calculation of the Influence of Factors on the Mean Annual Wage Indicator for 1971-1980

Indicator	1971	1980	Absolute Increment
Total work volume, millions of rubles--V	46.8	201.9	155.1
Average listed number of employees, thousands of persons--T	29.6	75.8	46.2
Mean annual wages per worker, rubles--W	1,584	2,665	1,081

1. Growth in work volume due to increase in number of personnel: $\Delta V_T = W_0(T_1-T_0)=1,584 \cdot 46.2=73,180.8$, or 73.2 million rubles, which is 42 percent of the increment of the work volume.

2. Increment in work volume due to growth in labor productivity: $\Delta V_W = T_1(W_1-W_0)=75.8 \cdot 1,081=81,939.8$, or 81.9 million rubles, which is 52.8 percent of the increment in work volume.

The influence of both factors--intensive and extensive--on the increment in work volume from 1971 to 1980 varies insignificantly. Thus 52.8 percent of the increment was attained in this period due to growth in labor productivity while 47.2 percent was attained due to an increase in the number of workers. However, further calculations show that the role of the intensive factor increased sharply in the 10th Five-Year Plan as compared to the 9th. In 1971--1975 the increment in work volume occurred mainly due to the extensive factor--that is, an increase in the number of workers. In that period, 78.8 percent of the increment in work volume was attained due to an increase in the number of personnel, while growth in labor productivity caused the work volume to increase by only 21.2 percent. In the 10th Five-Year Plan, on the other hand, the intensive factor--that is, growth in labor productivity--played the decisive role in increasing the work volume. It was responsible for 74.5 percent or almost three-fourths of the increase in work volume.

The proportion of the active part of fixed productive capital exhibited a low mean annual rate of growth--3.1 percent. The absolute increment of this indicator averaged 2.3 percent per year. The growth rate of the indicator was higher in the 9th Five-Year Plan than in the 10th, being 5.2 and 1.9 percent correspondingly.

The proportion of jobs carried out by an electronic computer in relation to the total job volume increased continuously. From 1971 to 1980 this indicator increased by more than fourfold. The growth rate of the proportion of jobs carried out by electronic computers was approximately the same for the 9th and 10th five-year plans, while as a whole the growth rate was 16.8 percent in 1971-1980. The mean annual absolute increment of the indicator during this period was 1.7 percent.

Beginning with the end of the 9th Five-Year Plan the computer system of the RSFSR TsSU began to be supplied with more-productive third-generation electronic computers, which naturally had an influence on the mean daily load on electronic computers. The mean absolute increment in this indicator was -0.1 hours in the entire period of analysis.

Wages calculated per man-hour of worked time attest to continual growth in the mean hourly pay of workers in the computer system of the RSFSR TsSU. In the period under examination, this indicator increased by almost 1.5 times--from 0.51 rubles to 0.72 rubles. This indicator is typified by a stable growth rate. It increased by 3.9 percent in 1971-1980.

Analysis of the dynamics behind wage outlays per ruble of completed work volume permitted the conclusion that the rate of growth of the work volume exceeded the rate of growth of wages in the computer centers of the RSFSR TsSU. This indicator steadily decreased at a mean annual rate of decline of 1.1 percent in the 9th and 1.7 percent in the 10th Five-Year Plan.

Outlays per ruble of completed work volume also exhibited a general tendency to decrease, but this occurred rather slowly and nonuniformly. The production costs of work decreased by a total of 3.1 percent in the entire period of

analysis; growth of this indicator, elicited mainly by introduction of new pay conditions, was observed in certain years.

We can see from Table 1 that although the computer system of the RSFSR TsSU is working profitably, stable growth in profitability is not observed. Nonetheless not only was the decrease in profitability which occurred in 1971-1975 halted in the 10th Five-Year Plan, but also an increase of 10.7 percent was achieved.

Thus analysis of the development of the computer system of the RSFSR TsSU shows that the basic efforts must be concentrated on the fastest possible development of electronic processing of statistical reports, on raising the load carried by electronic computers, on improving the organizational structure of the computer system, on improving the production processes used in computer data processing and on renewal of the computer pool at the rayon level.

11004
CSO: 1863/172

OPERATING EFFICIENCY OF COMPUTER CENTERS IN ESTONIA EVALUATED

Tallinn SOVETSKAYA ESTONIYA in Russian 11 Feb 83 p 2

[Article by T. Vel'dre, secretary of the section on automated control systems and computer hardware of the technical economic council of the Tallinn municipal committee of the Estonian Communist Party: "Where are Computers More Efficient?"]

[Text] Some two-thirds of the rise in labor productivity in Estonia is provided through the implementation of scientific and engineering achievements. The introduction of automated control systems and computer facilities plays its own part here also. The section devoted to these problems, which functions as part of the technical economic council with the Tallinn party municipal committee, together with the section on computer engineering of the Scientific and Engineering Society imeni A. Popov recently conducted a competitive review of computer centers. The purpose of it was to ascertain where computer equipment is being best introduced and more efficiently utilized and to make this experience available to others.

The competition commission awarded first place to the information computer center of the Estonian SSR State Committee on Television and Radio Broadcasting. The average annual output per worker here (about 7,000 rubles) is one of the highest in the republic. The output of useful computer time per worker is also one of the best. In other words, they know how to work economically and efficiently in this center. The percentage of computer time losses is low here; the computer makes use of the services of a centralized comprehensive service system. Additionally, approximately each 10th worker here is an efficiency expert .

Second and third places went to the computer center of the Planning and Design Office for Control System of the Estonian SSR Ministry of Light Industry and the computer center of the Estonian Union of Consumer Cooperatives respectively. The largest computer center, the repubic collective use computer center of the Estonian SSR Central Statistical Administration, which had garnered prize

places in previous competition, was inferior this time to smaller staffs in work efficiency and economy.

On an average for the republic today about six percent of useful computer time is lost because of technical defects. Time sharing is little used, that is, the possiblity of solving a large number of problem simultaneously on a computer. Some computer center continue to repeat the development of programs which are already available in the republic. For example, programs for calculating wages have been repeatedly developed more than 10 times.

For a better comparison of the results, the section on automated control systems and computer equipment with the party municipal committee is proposing the organization of a socialist competition among computer centers, set up on a cost accounting basis. The directors and secretaries of the party organizations of the centers are invited to the party municipal committee on February 16th at 3:00 PM to discuss issues of setting up such a competition.

8225
CSO:1863/184

NEED FOR NATIONAL ECONOMIC INFORMATION MANAGEMENT IMPROVEMENT UNDERSCORED

Moscow PRAVDA in Russian 2 Jul 83 p 3

[Article by Yu. Kanygin, doctor of the economic sciences and chief of the laboratory of the institute of cybernetics of the UkSSR Academy of Sciences, Kiev: "The Information Framework for Management. Intensification: the Essence, Ways and Means"]

[Text] No matter what part we take of the intensification program planned by the 26th Party Congress in the June Plenum (1983) of the CPSU Central Committee, its implementation entails an upsurge of information handling. Whether the issue is one of improving utilization of production capacities, accelerating scientific and engineering progress, a further increase in labor productivity, saving energy and materials or "long range" planning - all of this brings up more complex statistical and accounting, analytical, planning and organizational problems than in the past. Their resolution requires the processing of increasing volumes of diverse information.

About 600 billion management operations (official acts) are completed annually in our country, which are accompanied by the preparation and processing of approximately 60 billion administrative and planning documents (a 10 page manuscript is taken as the unit here).

The intensification which is producing the new information upsurge sharply presents the problems of an all-around improvement in the efficiency support for management. And it is important here to skillfully utilize the capabilities of electronic equipment. They cannot adapted to the requirement of the intensification by improving each component (accounting, plan indicators, financing procedure, prices, legal norms) either separately or within the scope of sectors and enterprises.

With the coming of new growth factors into first place (intensive factors related primarily to scientific and engineering progress) as well as new tasks (social tasks related to the optimization of the final results), the accounting system is need of a qualitative transformation. With the abundance of the data being recorded, including secondary data (at the average industrial enterprise, three to four million indicators are recorded), the incompleteness of the accounting is increasing. The accounting for intensification indicators is the most poorly formulated and procedurally

organized: scientific and engineering development indicators, indicators for the scientific technology input requirements of production as well as automated control system efficiency indicators. Intensification indicator accounting at the regional level is poorly handled. The existing forms of the central statistical administration provide too little of the information necessary for managing the development of regions.

The reliability of primary information is being reduced because of the fact that primary account is not always linked to bookkeeping, and the observance of estimates and standards. Strengthening the control of the ruble, and success in combating added costs are governed in many respects by the level of the bookkeeping accounting and its linkage to operational accounting.

Automated control systems can operate efficiently only with strict and reliable accounting for all of the important aspects and phenomena in economic management practice. The quality of the information product, which is produced by the computer hardware, depends in the final analysis on what we put into the computers. This makes it all the more necessary to improve accounting efficiency.

According to data from studies, some two to three times as many documents are prepared at enterprises as are needed for management. Only half of the information in reporting forms contain anything new on the work of the enterprises. The remainder is a repetition of plan and standards data, which are known to the recipient. This overloads and even fouls up the management information systems and gives birth to an inclination to inflate the administrative apparatus.

Experience shows that the number of documents can be significantly reduced by means of their unification and standardization, the use of displays as well as computer memories for recording and transmitting information. The list of such documents as reports on individual output, job authorizations, and consignment notes (there are hundreds of forms) can be reduced by a factor tens of times. Reasonable regulations for the quantity and types of documents are also needed.

However, in order for the work of simplifying office work and improving the efficiency of document turnover to be brought up to meet modern requirements, it is necessary to carry it through not only within the framework of individual departments, but also at the intersectoral level, and provide for coordination on a statewide scale. It is necessary to more widely expand work on the implementation of the scientific organization of labor in office work and coordinate this effort.

Economic indicators comprise the framework of the information system for management. They should be set up in a strict system, oriented towards the final results. It is planned in the current five-year plan that there will be a more extensive utilization of such quality indicators as the scientific input requirements for production and the scientific and technical potential of an enterprise, the social development of production collectives, etc. Work is underway on the standardization of procedures and setting norm and

reference data for calculating indicators in various sectors. Decisions of a systemwide nature are also needed here, which make it possible to link together the actions of planning and financial organs, central statistical administration, State Committee on Prices and other departments.

In specific terms, this means that it is necessary to set up a more sophisticated information base for plan administration, which is fully adapted to program oriented techniques of planning, optimization of decisions using computer modeling, etc. Program oriented techniques, just as large computer networks, cannot be simply hung on traditional organizational structures. Organs are needed here which are free of departmental blinders. And not just this. New indicators and evaluation criteria as well as a new financing procedure, and new (interactive) methods of arriving at administrative decisions are also required. For this reason, without disparaging the importance of work on putting the information flows in good order, which has been carried out within the bounds of individual departments, I will note the sharp necessity of improving interdepartmental interaction in this area.

The Central Statistical Administration, Gosplan, Ministry of Finance, Gosbank and State Committee on Prices establish the methods, lists, and forms for calculations of economic indicators, as well as the forms and contents of planning, accounting and reporting documents, the periodicity of their presentation, etc. Within each sector, this is assigned to the central apparatus of the appropriate ministry (or department). No matter how well these organs operate, at a certain stage in the increasing complexity of the information system, they come up against barriers. The State Committee on Prices naturally does not answer for national economic planning (Gosplan is for this), the Central Statistical Administration does not answer for prices and the Ministry of Finance is not occupied with the organization of operational accounting (finance accounting is important to it), etc. And all of this is not simply bits and pieces, but inseparable components of a single flexible system.

The information system for plan management has now become greatly more complex and more dynamic. In my opinion, it is necessary under these conditions for the USSR Council of Ministers to look to a special large scale service, which besides rights and obligations, would have a network of its own organizations and would systematically examine and work out comprehensive issues of improving management and its information base. Proposals to set up such a service (without drawing on additional personnel) have already been put forward. Thus, academician V. Glushkov proposed the creation of a USSR State Committee on Control Systems as an arm of the USSR Council of Ministers to improve the information foundation for plan management.

However, an improvement in administrative structures, indicators and forms of documents is important in and of itself, but will not completely resolve the problem if it remains in the traditional rut of "paper" technology which has come into being. Administrative organs, no matter how well they function, with increasing frequency do not have time to process the information. The

design of man-machine information systems has become not simply an element in the improvement of control efficiency, but an important quality of the planning system, and automated control systems do not play the part of appendages to the administrative apparatus, but are an integral part of the organizational process.

It is necessary to get a firm grip on the matter and merge the refinement of the economic management mechanism, the improvement of organizational structural efficiency, the systematization of information flows and the creation of computer networks together. This will make it possible to conduct an economic management policy which is rather flexible and sensitive to the winds of scientific and technical progress and complex conditions. For this reason, a universal comprehensive approach to the refinement of organizational processes, including the widescale utilization of computers, is contained in the resolutions of the 26th Party Congress.

It is important to get at the root of things; the greater the electronic computer potential, the more pronounced the defects in its utilization are manifest. Our computer centers are distributed among those sectors and areas which they service. They are far from being utilized at their full capacity. There are still very few collective use computer centers which employ time sharing in the solution of several various problems simultaneously. Overall supervision of computer use (including procedural) has been partially taken over by the USSR Gosplan, partially by the USSR State Committee on Science and Technology, partially by the USSR Central Statistical Administration, and other organs. The distribution of large computers among the sectors and enterprises is handled by Gosplan, small computers by Gossnab and punched cards, magnetic storage vehicles and perforation equipment by the Central Statistical Administration.

This is how things have historically come about. It was possible to be reconciled to this while the processes of the initial acquisition of computers were underway. But a certain critical mass already been achieved and a complex situation has arisen. And now, as it seems to me, it is impossible to be satisfied with the situation where computer information science is organizationally broken up and deprived of the advantages of centralized control. Working out the requirements and limitations for the expansion plans of the information industry, the distribution of capital investments and fundable resources (predominantly the electronic computer hardware itself) - it is thought that it would be worth it to turn all of this over to a single state organ responsible for a single technical policy and for acting as the single customer with respect to the ministries producing computer equipment. The State Committee on Control Systems could become such an organ.

The USSR State Committee on Science and Technology has at times been forced to take over the control of the increasingly complex managment of computer information science and has done no small amount in this field. However, it cannot be involved in current questions of control in various sectors of the national economy; its sphere is the direction of technical progress in them.

In such a matter as improving control, there is no place for independent action, giving priority to local interests or amateurish work. For success here, one must make provisions for the operationally timely interaction of numerous departments, organizations and enterprises. And it is difficult here to make do without a high level state organ responsible for the entire complex of measures related to this.

8225
CSO: 1863/184

DEVELOPMENT OF AUTOMATED SYSTEM OF PLAN CALCULATIONS

Moscow PLANOVOYE KHOZYAYSTVO in Russian No 5, May 83 pp 65-73

[Article by V. Bezrukov, deputy chief of GVTs Gosplan SSSR [USSR Gosplan Main Computer Center]]

[Text] Methods of mathematical economics and computers have been used more and more extensively in the last decade in planning the national economy. During this time, a great deal of work was performed on designing the automated system of plan calculations (ASPR) in the USSR and union republic state planning committees and local planning agencies. Developing the ASPR is considered a major direction in the comprehensive improvement of the methodological, informational and technical base for planning.

The ASPR is being developed by making traditional methods and planning technology more efficient based on modern achievements of economic science, consistently improving planning in the direction of developing a unified system of plans, and introducing modern methods for processing and displaying economic information which allow making efficient use of computers. This approach allows raising at each stage of ASPR development the quality of drafts of state plans for economic and social development and improving their comprehensiveness, interrelatedness and balance.

In developing the theoretical and practical base for the system, specifications were compiled for developing the ASPR as a whole and its functional and supporting subsystems at the level of the USSR Gosplan and the state planning committess in the union republics, Moscow and Leningrad. In them, specialists performed analysis of the traditional technology for drafting plans and formulated the main goals and tasks of improving the plan under the conditions of extensive use of the methods of mathematical economics and computers. Based on them, the draft ASPR was developed and approved and the technical designs for the functional subsystems were prepared and approved.

Given in these documents with regard to the complex of measures on improving planning is a flowchart of the plan process reflecting the logic and sequence of development of drafts of state plans and monitoring the course of their fulfillment. In general form, it is also tied to the schemes for functioning of sector management information systems for the ministries and departments which interact with the ASPR in the course of developing plans. Reflected in these schemes are the concrete results of scientific research in the field of improving techniques of long-term, five-year and annual planning. Considered within the draft ASPR and solved in the complex are the fundamental problems of creating the informational,

hardware, software, technological, personnel and legal organizational support for the process of developing drafts of state plans and monitoring the course of their fulfillment.

As a whole, the complex of documentation is a reliable base for systematic improvement of the technology of planning under the conditions of extensive employment of methods of mathematical economics and computers.

Much work has been performed in organizing ASPR design and introduction. Now taking part in developing it are over 100 scientific research and design organizations under the USSR Gosplan, the union republic state planning committees, the ministries, departments and USSR Academy of Sciences. The activity of these organizations is prescribed by the coordination plan for efforts which is approved for a five-year period by the USSR GKNT [State Committee for Science and Technology] upon submission by the USSR Gosplan. Defined in it is the list of main tasks for ASPR development and schedules for implementing them in the given period and the financial resources needed for this are identified. Based on the plan, the co-executive organizations formulate concrete working programs which together with the suggestions from the state planning committees in the union republics, Moscow and Leningrad are used as the basis for preparing and approving the annual plans for ASPR introduction. Corresponding methodological materials are prepared to ensure unity in forming the system of plan documents. In the organizational aspect, the role of departments in the USSR Gosplan, union republic state planning committees and local plan agencies in introducing ASPR methods and facilities is growing substantially. ASPR lead client functions are exercised by the department for improvement of planning and economic incentives in the USSR Gosplan; formation of working groups in the USSR and union republic state planning committees has been completed. They are charged with the responsibility for operational guidance of design and organization for introducing ASPR functional subsystems; the regulation on working groups has been approved. To strengthen methodological and organizational direction of their effort, it is necessary to activate the work of the USSR Gosplan Coordinating Council on Matters of ASPR Development.

To enhance personal responsibility for quality of design of the ASPR as a whole and its subsystems, the functions, rights and duties of the chief director of efforts on ASPR development have been precisely specified. Some changes have to be made to the staff of his deputies and lead directors of efforts on functional subsystem design and development. To improve ASPR development coordination at the republic level, regulations were approved on the lead republics responsible for comprehensive design for republics with the oblast subordination (Ukrainian SSR), without oblasts (Lithuanian SSR), and for comprehensive solution to the problems of ASPR interaction with ministry and department sector management information systems at the republic level (Latvian SSR); the staff of lead directors of efforts on ASPR development at the republic level has been precisely specified; formation of inter-republic working groups on functional subsystems is essentially complete. This organization allows ensuring the required level of responsibility and results in ASPR design and introduction.

In accordance with the unified plan formulated by the USSR Gosplan Main Computer Center, development of the ASPR hardware base is underway.

In the area of ASPR information support, specialists have developed and are assimilating in stages the all-union classifiers of sectors in the national economy, industrial and agricultural products, institutions and organizations, efforts and services, technical-economic indicators, system-wide classifiers and special-purpose nomenclatures for the ASPR. In operation at the USSR Gosplan Main Computer Center is a common information bank of numeric and text information; this bank is used in data entry, accumulation, storage and updating operations.

Facilities supporting operation of the information bank have been developed and are being introduced within the framework of ASPR system-wide software.

Much has done in the area of personnel training and retraining. Thus, in the Higher Economic Courses under the USSR Gosplan, where USSR and union republic state planning committee employees are trained along with ASPR developers, the department for "Application of Methods of Mathematical Economics and Computers in Planning" has been established; it provides methodological direction for the training process; programs with regard to specialization of courses have been developed; methodological texts are produced regularly, etc.

ASPR development efficiency is largely governed by the development and introduction of economic plan problems solvable by using computers in generating drafts of state plans for economic and social development.

About 12,000 problems used directly in developing essentially all sections of state plans have been put into operation in the USSR and union republic state planning committees and in the Moscow and Leningrad city planning commissions. Thus, about 4,000 problems were being handled by computers in the USSR Gosplan in 1982. Republic level ASPR's are being developed at a fast rate. Deserving special mention here is the great effort performed by the RSFSR Gosplan; just in 1982 alone, about 300 problems were placed into operation. The majority of them are run on minicomputers. Much progress was made in 1982 by the UkSSR Gosplan and the KaSSR Gosplan which introduced more than 170 and about 70 problems, respectively, etc. Efficient use of these problems in planning technology is substantially enriching the analytic arsenal of the plan employee through processing of large files of plan and accounting information, the many alternatives in performing computations and optimization of plan solutions.

An example is the analytic and plan computations made on the base of consolidated intersector balances.

An information base has now been formed at the USSR Gosplan Main Computer Center. It contains a dynamic series of indicators for 20 years as part of the summary intersector balances on a profile of 18 sectors of intersector production. This information is the necessary basis for multivariant calculations for drafts of five-year and long-term plans to assess the various hypotheses for socioeconomic development and study the effect of planned indicators of efficiency of the individual sectors on the growth of social production and meeting final social demand.

Another example is using models of optimization to assess development and siting of complexes of sectors, individual sectors and types of production to determine the best specialization of enterprises and possible alternatives for modernization of existing enterprises and construction of new ones to meet national economic demands

for corresponding products with the least outlays. Thus, computations are made at the USSR Gosplan Main Computer Center for the future period for the chemical, petrochemical, lumber, wood-working, light, and food industry, agriculture and forestry, and for a number of other sectors. But development of these calculations at the national economic level is being held back on the part of certain ministries that in a number of cases are trying to obtain some more resources at the stage of development of plan drafts while understating their capabilities for development of production. Here, the mutual interests of the parties in the ultimate national economic effect must be ensured through appropriate incentives for economic results of production.

Much work has been accomplished in generating norms for physical resource consumption which has allowed ordering the existing normative base and more profound analysis of norm-forming factors.

Performing calculations on this basis allows making extensive use of the advantages of the normative method of planning in defining demand for physical resources and substantiating plans for production and capital construction, ensuring in the process enhancement of plan balance.

Plan calculations performed on computers cover the basic stages of development of long-term and current plans and include the following calculations: on analysis and monitoring the course of implementation of plans, and determining the expected fulfillment of them; for substantiating indicators of drafts of plans with regard to specified limits of capital investment, equipment and physical resources; consolidated-analytic, which enable analysis of drafts of plans submitted to USSR Gosplan by ministries, departments and union republics, as well as calculations allowing receipt of finished plan documents in machine form.

But the main share of problems handled within the ASPR still operates in the autonomous mode and data exchange between them is limited; this substantially reduces computer utilization efficiency in plan work. This is due, on the one hand, to the effort on ASPR development being oriented until now directly to the needs of USSR Gosplan separate departments without consideration of their interrelations in the process of developing plan drafts, and on the other, to inadequate development of computer hardware produced by domestic industry.

Decisions by the 26th CPSU Congress stipulated introduction and efficient use of the ASPR in the 11th Five-Year Plan, which predetermines the high demands for accomplishing the complex of efforts on establishing and developing it. They are being implemented in the following directions:

development and introduction, for annual planning, of information and computing complexes on balancing calculations of indicators of plans for production and capital construction and furnishing the resources required for them;

design and introduction, for five-year planning, of a central complex of problems to substantiate basic indicators in plans for economic and social development;

for purposes of long-term planning, development of an integrated complex of balance calculations to substantiate basic indicators in plans for USSR economic and social development and to monitor the course of fulfillment of state plans, information and computing complexes; and

development of computerized information systems for monitoring the course of fulfillment of plans and for the basic directions of plan work.

Within the scope of the first direction, a complex of balance calculations of indicators for development of physical production will be implemented by using a progressive normative base.

The complex of balance calculations of indicators for development of physical production includes the problems, previously introduced and those being newly developed within the scope of the USSR Gosplan ASPR, for the consolidated subsystems "Physical Balances and Distribution Plans" and "Standards and Norms" as well as the sector subsystems in the USSR Gosplan ASPR and combines the calculations for generating and maintaining the normative base for planning the supply of materials and equipment for basic production, capital construction and repair-operating needs, calculations on compiling physical balances, plans for distribution of resources and extracts from them, and determining actual use of physical resources for the reporting year.

Calculations are combined into a complex in accordance with the technology of drafting the annual plan represented in the ASPR conceptual design in the principle scheme for the unified system of plans and made more concrete in the detailed designs for the functional subsystems in the form of functional-structural schemes for drafting annual plans.

Implementation of these calculations on computers will allow:
making more detailed and refining the annual quotas of the five-year plan and the resources required to meet them with regard to the course of fulfilling the five-year plan as well as the changed conditions and capabilities; a comprehensive approach to planning and use of various types of material-technical resources to most fully meet the demands of the national economy for the current period; integration of balance calculations and a closer tie-in to sector, territorial and address sections of the annual plan; use in them of the progressive normative base which considers the impact of scientific and technical measures in social production; intensification of monitoring implementation of annual plans for supply of materials and equipment, timely identification of additional reserves, and analysis of deviations from plan quotas; enhancing substantiation of plan solutions by raising the accuracy of calculations, expanding the range of indicators considered, and analyzing the quality of information used; creation of a common information base to achieve methodological unity in performing plan and analytic calculations.

A fundamentally new and major direction in the development of ASPR efforts is the development and introduction of the central complex of tasks (TsKZ) intended for multivariant analysis of the basic indicators for the country's economic and social development, solving general economic and intersector balance problems and raising the efficiency of social production, and substantiating rates and proportions of development of the national economy in the process of working up drafts of the five-year plan.

This complex includes calculations for balance of the national economy, consolidated cost and expanded real-cost intersector balances, balances of fixed assets, production capacities and capital investment, financial and labor resources, basic

indicators for the functional sections in a plan (standard of living, production cost and profit, foreign economic ties and others), as well as key indicators in sector plans. The sequence of plan calculations being consolidated into the complex is governed by the technology for drafting the five-year plan.

Functioning of the central complex of tasks must support:

in accordance with socioeconomic aims in a given plan period, generation of alternatives for rates, general economic and consolidated intersector proportions of development of the national economy ties to previously specified limits of basic national economic resources and efficiency of their use based on the capabilities of scientific and technical progress;

in accordance with the target settings and indicators obtained in the course of implementing the first group of functions, defining the indicators for the production plan and the resources needed for it by the sectors of physical production and indicators by types of activity and demand for resources by the sectors in the nonproduction sphere, and generation of the basic indicators and balances by the functional and resource sections in the plan; and

analysis, assessment from national economic positions, matching and balancing of indicators derived as a result of implementation of the second group of functions, and generation of the interrelated system of basic indicators for the country's economic and social development. Hence, it is evident that development and introduction of the central complex of tasks is a very critical and laborious matter covering the effort on the five-year plan of the majority of USSR Gosplan sector, functional and balance departments.

Very complex problems within the scope of ASPR development have to be solved to get efforts underway to develop an integrated complex of balance calculations to substantiate the indicators in the plans for USSR economic and social development in long-term planning. Here is just a small list of the problems that have to be solved in the long-term planning process: defining methods to directly reflect in plan calculations the effect from implementation of the achievements of scientific and technical progress, use of new types of raw materials and energy, introduction of fundamentally new technologies, the effect on rates and proportion of economic and social development, assimilation of new territories and training of highly skilled personnel.

In accordance with this, the complex of models and balance calculations must be oriented to generating indicators that describe the solution to major economic and social problems and their relation to indicators of scientific and technical progress, and to substantiating on this basis the intersector aspects of the investment strategy and alternatives of rates and proportions of development of the national economy in the future long-term period.

Demographic calculations to 2015 are now being made, and prospects for development of the network of children's preschool institutions, secondary specialized and higher education and a number of others are being defined. But to develop the complex of calculations, we have to substantially expand the introduction of forecasting methods within the scope of the ASPR, use simulation modeling methods and construct socioeconomic scenarios.

Developing the integrated complex of balance calculations which reflect both the physical-material and financial aspects of social reproduction will allow expanding the range of factors considered in drafting alternatives for national economic dynamics and the structural policy in long-term planning.

In addition to these complexes, a major aspect in ASPR development is the complex of tasks called the Unified System of Capital Construction Planning (YeSPKS). Its introduction is called upon to facilitate elimination of the substantial shortcomings in planning the investment process that are due to the lack of the needed level of comprehensiveness in planning of capital construction as a unified process covering the planning and distribution of capital investment, generating of title lists, planning of project surveying operations, construction and installation work, and introduction of facilities and capacities that support optimal capabilities for expanded reproduction.

In accordance with this, the Unified System of Capital Construction Planning as part of the Automated System of Plan Calculations is oriented to solving the following problems:

achieving coordination in planning development of production and capital construction and achieving a direct link between the plan allocation of investment and the production program being planned;

achieving unity between the phases and cycles of capital construction, including planning of capital investment, project surveying operations, construction and assimilation of capacities;

enhancing the degree of balance of plan quotas by amounts and schedules for capital construction with the labor, material and financial resources needed to accomplish it;

consistent shift of planning of capital construction and accomplishment of it to a normative basis, keeping in mind adherence to normative schedules for construction and introduction of facilities, normative amounts of uncompleted construction and norms for costs of labor and physical resources;

selection of plan solutions that allow implementing the construction program at the highest possible level of efficiency of capital investment by making extensive use of the achievements of scientific and technical progress in capital construction;

and providing for the functions of monitoring the course of development and implementation of capital construction plans and coordination of activity of all main participants in the investment process during compilation and fulfillment of plan quotas.

The Unified System of Capital Construction Planning combines the operation of the composite resource-balance subsystem "Capital Investment," the composite-functional subsystem "Project Surveying Operations" and the sector subsystem "Construction and the Construction Industry." Also provided for within the scope of this Unified System is close interaction between these subsystems and the subsystem "Composite National Economic Plan," in which are generated the specifying and limiting indicators for capital construction, and with the latter, coordination of the plans is achieved on output of products and insertion and removal of fixed assets and production capacities.

The Unified System of Capital Construction Planning is being developed for all modes of planning (annual, five-year and long-term) in two phases. In the first phase within the scope of the 11th Five-Year Plan, we will experimentally test out the machine technology for processing information contained in the unified construction project cards that will come from ministries and departments.

In the second phase, the interactive mode of operation will be implemented in the process of drafting capital construction plans as we solve the organizational problems related to regularity of updating data bases created at all stages of plan work and in the course of developing reliable hardware for system operation.

Monitoring fulfillment of state plans is a major function in plan work. The aim in developing computerized information systems to monitor completion of state plans is to generate information to work out steps to eliminate bottlenecks occurring during plan fulfillment, to identify additional internal reserves and draw them into economic turnover, and on this basis, achieve balance in plan indicators. Within the scope of the ASPR, these calculation complexes must be in all functional subsystems and include in them the problem related to analysis of reporting data and determining the value of their deviations from plan quotas, assessment of possible impacts from these deviations on the future course of plan fulfillment, reflection and analysis of the reasons for deviations of the actual course of plan fulfillment from plan quotas.

In the appropriate subsystems, such complexes are now being developed primarily to monitor fulfillment of plan quotas by general economic indicators and major indicators that describe growth in labor productivity, increase in capital investment effectiveness, startup of production capacities, reduction in cost of physical resources per unit of product made and work, structure of production costs and a number of others. The main source of initial information on plan fulfillment is the statistical report that comes into the ASPR from the ASGS TsSU SSSR [USSR Central Statistical Administration Automated System of State Statistics].

Another direction in ASPR development efforts in the 11th Five-Year Plan is the development of reference information systems on the basic directions in organizing plan work. They will be designed to form corresponding information banks by USSR Gosplan administrations. Preparation of such a system has now begun for the Administration of Trade Turnover and Trade Resources. Experience from developing it will allow formulating specific requirements for the structure of the information bank, mode of updating it and frequency of access to the bank, types of queries and methods of generating them, etc. At the same time, we will also develop further the automated system "Personnel," monitoring executive activity of the USSR Gosplan apparatus, clerical work, system for monitoring drafting of the state plan and others.

Much work on ASPR development will be performed in union republic state planning committees. It will be in the same directions as in the USSR Gosplan, but oriented to the specifics of developing the economy in each republic as well as to the level of efforts reached in ASPR development.

To enhance the efficiency of introducing the ASPR into plan administration practice, we have to ensure its interaction with ministerial, departmental and republic management information systems [MIS].

Within the scope of approved programs of efforts, we are now solving the problems of interaction with joint operation of the USSR Gosplan Automated System of Plan Calculations and USSR Central Statistical Administration Automated System of State Statistics, the USSR State Committee for Supply of Materials and Equipment MIS, the Ministry of Foreign Trade MIS, the State Committee for Foreign Economic Relations MIS, and the sector MIS's for the USSR Ministry of Instrument Making, Automation Equipment and Control Systems and the USSR Ministry of Light Industry. The program of efforts with the USSR Ministry of the Petroleum Industry is the coordination and approval stage.

To solve interaction problems, we also have to interface the complexes of tasks performed in the USSR Gosplan ASPR with the union republic gosplan ASPR's. The initial interface will be between the USSR Gosplan Main Computer Center and the Computer Centers for Gosplans in the UkSSR, LiSSR and LaSSR with subsequent dissemination of results obtained to the other republics through appropriate functional subsystems by coordinating the composition of plan tasks at various levels of planning. Plans here call for the USSR Gosplan to send through its Main Computer Center to union republic gosplans the indicators for the list of construction projects to be accomplished within their territory, amounts of project surveying work for future construction, siting of industry for products, amounts of output of physical resources and demand for them, population count, migration, the breakdown by sex and age, composition of families, and major indicators for the sector plan. In turn, the union republic gosplans will send to USSR Gosplan on machine media the drafts of plans for comprehensive economic and social development of the union republic for their consideration, coordination and drafting on this base of the consolidated indicators in the state plan on a territorial profile.

These directions of efforts can be efficiently implemented by phased development of a distributed information processing network that allows plan employees direct access to data bases stored on computers to enter, update and process economic plan information while generating plan drafts.

ASPR hardware is now going through the stage of re-equipping with new domestic hardware. Computers with higher performance and better suitability to plan calculations are being assimilated: the YeS-1045 with a speed on the order of 0.5 million operations per second and equipped with disk storage units holding 100-200M bytes. Plans call for acquisition and installation of data teleprocessing processors to support operation of the computer system in the USSR Gosplan Main Computer Center in the computer network at the top level of the ASPR, and connection of terminals in the USSR Gosplan and organizations under the USSR Gosplan to the computer system in the USSR Gosplan Main Computer center. A new higher performance system for large-scale data entry and preparation is being assimilated. Unified Computer System video terminals are being installed in the USSR and union republic gosplans. Assimilation of the "Iskra-226" minicomputer, developed especially for its extensive use by plan employees as a means for organizing specialized data bases and on-line information processing, has begun in subdivisions of the USSR and union republic gosplans. Efforts are underway on developing system-wide software, including support for the "Iskra-226" minicomputers and facilities for organizing data exchange between computers.

Completion of these efforts will allow combining the individual high performance Unified System computers into computer complexes by using special hardware and software to efficiently site the centralized information bank and software on several machines, and connecting to a given complex the minicomputers as intelligent terminals, which will allow organizing with them maintenance of specialized data bases for the needs of a specific user and tying them to the centralized bank to solve major economic plan problems requiring large amounts of information.

Development of the computer network will allow:

offering the plan employee the needed computer capacities in the process of drafting and producing plans and generating all economic plan docuemntation possible by ensuring direct access to the ASPR information bank by using the minicomputers and terminals located at his work station;

significantly increasing the efficiency of work in the USSR and union republic gosplan subdivisions through enhancing the quality and efficiency of information acquisition, accumulation, updating, storage and processing; and

enabling an increase in reliability and efficiency of hardware operation through its redundancy and automated selection of the most efficient technology for solving the complexes of problems on the network.

On the whole, completion of these efforts on ASPR development will allow raising the efficiency of introducing methods of mathematical economics and computers into the practice of plan work and ensuring successful implementation of the decisions made at the 26th CPSU Congress.

8545
CSO: 1863/167

COMPUTER SUPPORT FOR BANKING OPERATIONS IN KAZAKHSTAN

Alma-Ata NARODNOYE KHOZYAYSTVO KAZAKHSTANA in Russian No 5, May 83 pp 50-53

[Article by candidate of the economic sciences M. Geydarov, chief of the computer center of the USSR Gosbank Kazakh Republic Office: "Computers in the Service of Bank Workers"]

[Text] There is a continuous growth in the volume of financial, credit, accounting, cash and other transactions in the institutions of Gosbank [State Bank] and Stroybank [All-Union Bank for Financing Capital Investments]. The rapid, high quality processing of the enormous flow of information related to these transactions can be accomplished now only by means of computers. The computer center of the Kazakh republic office of Gosbank was placed in service for this purpose at the end of 1974.

It started its activity servicing only three institutions in the system. In 1975, they already numbered nine, while the set of "Gosbank Operating Day" tasks was supplemented with the tasks of compiling the monthly reports (forms 721, 722 and 759) and the periodic reports on the execution of the cash transaction plan. The "Local Interbranch Turnover Receipt" task was implemented and then the "Comprehensive MFO [Interbranch Turnover] Receipt" task was introduced.

The necessity of expanding the circle of serviced institutions required the introduction of a remote data transmission mode using communications channels, and the use of punched tapes as an intermediate data storage vehicle. Following the appropriate technical preparation for this method of servicing, an additional four institutions of the system were incorporated within one year, while the set of "Gosbank Operating Day" tasks were supplemented with yet another six tasks, including a task developed by the specialists of our computer center: "Compilation of Records to Credit Payments for Bank Collection of Money Receipts".

It was specifically the computer center of the Kazakh republic office of Gosbank which was the first in the system to completely take in all of the Gosbank institutions within an administrative region (the Alma-Ata oblast) for data processing using computers. In the near future, it is planned that all of the Gosbank institutions in the Taldy-Kurganskaya oblast and then the Dzhambulskaya oblast will be incorporated also.

At the present time, we carry out the comprehensive "Operating Day" and "Interbranch Turnover Receipt" tasks using computers for 34 institutions in the three oblast offices of Gosbank. We also perform the "Interbranch Turnover Monitoring" task for all 311 branches of the Kazakhstan Gosbank, as well as for the 105 institutions of the Tadzhik and Turkmen Gosbank offices.

The positive results of servicing the Gosbank institutions of the three republics cited above with the "interbranch turnover monitoring" task have provided a basis for the central bookkeeping of the USSR Gosbank managing board to pose the question of the expediency of transfering the performance of this task to the computer center for the Gosbank institutions in Uzbekistan. Our specialists came up with a reworked task for the purpose of providing data processing for 680 institutions as well as for 50 offices of Gosbank.

This modification by the Main Computer Center of the USSR Gosbank was recently recommended for use by other computer centers based on the "BANK" computer complex, and for the computer center of the Kazakh Gosbank office, it opens up the possibility of forwarding the results of this task to all USSR Gosbank institutions in Kazakhstan and the Central Asian republics.

Briefly stated, the essence of the tasks which were the topic of the above discussion consists in the fact that they provide the capability of both analytical and synthetic accounting for balance sheet accounts, and in the near future, nonbalance sheet accounts as well, with the derivation of the corresponding registers; they also provide for monitoring interbranch turnovers in the Gosbank system, the performance of economic developmental work and the set of bookkeeping operations which arise when receipting reply interbranch turnovers, as well as manage the statistical reporting and the transfer of pensions.

It needs to be underscored that under conditions of computer use, one of the fundamental features of bookkeeping is the automated composition of a series of bookkeeping entries. In accordance with the specified program, the computer credits the sums to some accounts and removes them from others, and also makes bookkeeping entries when establishing the credit balance for loan accounts, distributing income between union and republic budgets, closing accounts in accordance with mutual settlements between Gosbank institutions, which are serviced by the computer center, etc.

As a result of automating the banking operations, it has become possible to completely free the staff from the compilation of the daily bookkeeping journal. Moreover, the computer cleanly and errorlessly checks the equality of the turnovers and residues in personal accounts and the balance sheet, and for this reason, has eliminated the necessity of maintaining summary cards on balance accounts. The use of computers and program techniques of monitoring in all stages of information conversion has provided for high precision of the calculations and eliminated the necessity of comparing entries in personal accounts with the documents on which they were based.

Significant positive changes have also taken place in the organization of the work of accounting for initial interbranch turnovers and receipting reply interbranch turnovers. While under usual conditions the reconciling of the initial and reply letters of advice was accomplished by the workers of the banking institutions, it is now done automatically by the computer. Clearing payment turnover and concluding turnover accounting is handled in a clear cut manner and incomparably faster with computers, and the data needed to compile reports, etc., are also acquired.

The following fact attests to the efficiency of the automation of banking operations. For the 17 Gosbank institutions in the Alma-Ata oblast in 1982, as compared to the outset of their acceptance of services by our computer center, the number of processed documents increased by 65 percent while the number of accounting operations staff workers was reduced by 15 percent. The picture is similar for the Gosbank institutions of the Taldy-Kurganskaya oblast, where with a 5.3 percent growth in the quantity of processed documentation, a reduction in the number of staff accounting workers of more than 14 percent was achieved.

Questions of boosting the labor productivity of workers engaged in computer operation at the computer center itself are constantly being examined. In the past year, we successfully increased this indicator by 13 percent over that of 1981. The readiness factor of the computer complex (the "BANK" computer complex) was brought up to 0.994; information has come to be processed in an operationally more timely manner with a higher level of quality and is also forwarded faster.

The average daily load on the computer complex over the last five years has increased by a factor of 1.7 times. The design capacity coefficient of the computers is increasing unabatedly. The plan assignments to reduce the expenditures and production cost of operations are being systematically fulfilled and overfulfilled. The consistent expansion of the group of tasks being carried out and the increase in the number of institutions being serviced are assisting us in achieving such indicators.

It must be noted that our computer center was one of the first to come up against the necessity of forwarding the surpluses of personal accounts at the beginning of the working day to distant Gosbank institutions. At first, the surpluses were forwarded manually by the telegraph operator in accordance with the current clearing accounts. With the operational start of the U-536 equipment and the mastery of the "Punched Tape Output of the Verification Control Sheet for Personal Account Surpluses" task, the situation fundamentally changed: the surpluses were transmitted from a punched tape, prepared by the computer. As a result, it became possible to process data from distant institutions (in particular, the Dzhambul oblast office of Gosbank).

For the purpose of consistently expanding the group of serviced institutions, another five Gosbank institutions were accepted for processing in 1980 and the "'Operating Day' Data Input via Communications Channels" task was implemented at the same time.

The transition to the new data input mode was preceded by a changeover to new formats for data input from punched tape.

After doing considerable debugging, it was possible to supplement the set of "Gosbank Operating Day" tasks with several more. Among them, the most significant is "Handling Final Turnovers for Emissions Operations".

The successful changeover of all of the serviced institutions to the progressive method of direct data input into the computers has had an extremely positive effect both on the activity of the operational accounting personnel and the specialists of the computer center. This has made it possible to significantly reduce, and then eliminate the previously existing stress in the work of the operators and staff members of the control groups for the Gosbank institutions being serviced, as well as the specialists of the production department for the reception, processing, checking and output of computer center data. Additionally, expenditures of material and labor resources for data transmission, processing and verification have been reduced.

The "'Operating Day' Information Output via Communications Channels" task, which provides for the transmission of personal account surpluses in accordance with form No.15 in an automated mode with the checking of the transmitted data via feedback was implemented to assure the timely output and reliability of information transmitted back to remote Gosbank institutions.

At the same time, the "Compilation of the Reporting of Financing of Major Repairs and Utilization of the Assets of the Production Expansion Fund in Accordance with Forms 749 and 849" task was introduced at the same time, and thereafter, a separate fifth task: "Statistical Reporting and Composing Economic Draft Plans on Its Basis".

The question of the necessity of operationally timely information feedback for the set of "Gosbank Operating Day" tasks to the most remote departments is becoming increasingly urgent in Kazakhstan and even in other republics. Computer center specialists have proposed the transmission of return information via a telephone channel following the reaching of their decision. But since the YeS-8010 data transmission equipment was not designed for operation with the "Ural 14-D" computer, the U-685, U-686 and U-546 interfaces were designed and manufactured for interfacing to the YeS-8010 equipment via a S-2 junction, and a circuit has also been proposed for synchronizing the U-685 interface to the YeS-8010 modem. The transmission rate has been increased in this case in the U-685 interface from 200 bauds up to 2,400 bauds. A number of other technical changes have been made and other agreements reached.

This method of high speed return data transmission has been successfully tested in accordance with a test program for Alma-Ata--Dzhambul communications. Thus, the technical aspect of the new equipment has been worked out. It is only necessary to resolve the question of software, concerning which we have forwarded specific proposals to the USSR Gosbank Main Computer Center.

And here is another example of increased efficiency from computer utilization. At their own initiative and under the supervision of the chief engineer of the computer center, L.N. Shchepin, and the chief of the communications facilities department, A.P. Bobrov, the specialists of this department installed in the computer center, as well as in the Dzhambul and Taldy-Kurgan Goasbank offices in a comparatively short time the TT48 multiplex equipment and exchanged data via a long distance telephone channel. This made it possible to save 235,300 rubles annually by eliminating the previously occupied telegraph channels.

An effective innovation to provide distant Gosbank institutions with information on personal account surpluses is the reworked "'Operating Day' Data Input via Communications Channels" task which was implemented in 1982 under the supervision and with the direct participation of the deputy chief of the computer center, N.K. Isin, and the chief of the computer equipment department, A.I. Ablezov. The operators of banking institutions now have the capability during the transmission of information to the computer center to query upon special request the residual amount in a specified personal account.

Along with the consistent expansion and refinement of automation of operations in the computer center, attention is being devoted to questions of mechanizing banking operations for the administrations of the Gosbank republic office. Thus, the total number of mechanized operations at the republic level has been brought up to 170 for this purpose. Among them are the following tasks: "Compilation of Dynamic Tables for the Expenditure of Wage Funds", "Execution of the Credit Plan", "Information on the Execution of the Cash Plan with a Rising Total From the Start of the Quarter (or Year) in a Cross-Section of Areas and by Items".

In the current year, we plan to start the automation of a number of banking operations at the republic level using the domestically produced SM-4 minicomputer which has provided itself quite well. The following will be accomplished in this way: a further expansion in the group of mechanization and automation operations; stress in the execution of work by reporting dates will be eliminated; it will become possible to obtain the processed data in various steps in an operationally timely manner; duplication will be eliminated in the selection of information; Economists will be able by working directly at a video terminal to do more complex jobs related to economic analysis; and it will be possible to make a transition to automation of the production processes in the apparatus of the Gosbank republic office.

There are a number of other reserves for boosting the efficiency of computer use. Thus, machine time from 8:30 to 17:30 hours is set aside for data input in our computer center. During this period, Gosbank institutions have the capability of transmitting information to the computers. At the same time, operation of the "BANK" computer complex has revealed that the transmission of the full amount of return information via low speed channels requires approximately the same amount of time. In this case, more than two-thirds of the computer time will be expended for data input and output. This can hardly be considered a normal phenomenon.

It is thought though that to subsequently use higher speed channels for each Gosbank division to change the ratio of the forward and return transmission times will not be justified.

It would be expedient and economically advantageous in the accounting operations of the bank to consider the questions of the possibility of curtailing the volume of information transmitted back without detriment to the actual demand for it. Thus, it is hardly necessary to transmit back to the Gosbank divisions all of the output of the computer center with respect to the entire products list and in the full amount, especially when all Gosbank institutions are serviced within the framework of an entire oblast and a summary balance sheet is issued.

An even more tangible reserve for boosting computer efficiency is found in the conversion of the largest computer centers, under appropriate conditions, to multiple user computer centers (KVTs) and collective use computer centers (VTsKP).

A new high performance computer center of the Kazakh Gosbank republic office is under construction at the present time in Alma-Ata and will be brought on line in the near future. It will be equipped with YeS-1045 computers capable of performing up to one million operations per second, which opens up extensive possibilities for further improving the control of banking and financial operations.

Considering the planned capacity of the new computer center, it will be possible in it to provide for data processing not only for Gosbank institutions, but also for the Stroybank and the Kazakh SSR Ministry of Finance.

8225
CSO: 1863/182

COMPREHENSIVE SCIENTIFIC RESEARCH AND DEVELOPMENT MANAGEMENT SYSTEM

Minsk PROMYSHLENNOST' BELORUSSII in Russian No 3, Mar 83 pp 67-69

[Article by G. Lopato, director of the Scientific Research Institute for Computers: "The Heart of the System is the Data Bank"]

[Text] A comprehensive work quality and scientific efficiency control system (KS UEN i KR) was developed and introduced several years ago in the NII EVM [Scientific Research Institute for Computers]. Encompassing all structural subdivisions and combining the principles of functional and specific program planning and control, it provides for high development work quality with a significant reduction in expenditures. The comprehensive work quality and scientific efficiency control system also determines the most important directions for the refinement of control processes, sets regulations for their technology, and provides for unity in the control processes, including that of planning the development of science and engineering, the volume of the development work and its level of quality as well as capital construction and reconstruction; it also produces a stable rate of increase in the efficiency of developmental work as well as an improvement in work quality. It assures the coordination of the resolution of large scale multifaceted problems, as well as the management of the cycle of developmental work in the stages of research, planning and fabrication of prototypes.

The comprehensive work quality and scientific efficiency control system consists of 10 main and 3 support subsystems. The most important of them is the automated design system (SAPR). And this is understandable since the major trend in computer equipment design is the qualitative improvement of the level of its development, production and applications through the use of new architectural solutions, sophistication of the hardware and production process base as well as searching out efficient forms and techniques for man-machine interaction. No less important factors are the functional and structural complexity of the computer systems being designed, the large number of diverse components with

complex interactions between them, and the increased requirements placed on equipment reliability and software as well as elevated requirements placed on the compatability of hardware and programs. Moreover, the rising complexity of computer architecture extends the development timeframes, increases the volume of material expenditures and requires drawing on large staffs of highly skilled specialists.

The problem of boosting the quality of developmental work in the design of modern computer systems is thus becoming the major one in the work of the planning organizations. A special role is set aside in the solution of this problem for unification: one of the main techniques of standardization. The multiplicity of products and processes has been successfully curtailed down to the expedient minimum, and their technical economic indicators have also been significantly improved.

Standardization and unification of computer hardware is broken down into two main areas: the standardization of the hardware from the standpoint of using unified and standardized products and components, basic support structures, sets of modules, panels, frames, etc.; and the standardization of the design documents.

The systematization and specification of the requirements placed on the hardware (TS) of the YeS EVM [unified system of electronic computers] and the expansion of design automation have necessitated the development of standardized structural designs not for individual computers, but for an entire generation of computers. For this purpose, a standard base support structure (BNK) was designed on the basis of experience from previous developmental work, and the standardized design solutions for this basic support structure made it possible to set up the series production of the YeS-1035, YeS-1060 computers and a number of individual pieces of hardware: the YeS-5566, YeS-3266, etc. The structural design standardization of the hardware base for the "Ryad-2" unified system of computers was determined in this way.

Standardized structural designs used in the YeS-1035 and YeS-1060 computers are formated in the form of base drawings and are disseminated among the related enterprises of the sector, something which provides for the standardization of development and manufacturing. Thus, the Minsk computer equipment production association has at the present time set up the manufacturing of basic assemblies and parts and is supplying them to the nation's other enterprises. The basic support structure of the "Ryad-2" YeS EVM has become that foundation on which the development of the structural design for the next generation of YeS EVM computers is based. By using this, one can also organize the specialized production of standardized parts and assemblies for the YeS EVM.

It is well known that the design of modern computers within curtailed timeframes is impossible without the use of special systems for the automation of the system level and engineering design by virtue of the complexity of modern computers, which contain hundreds of thousands of components. For this reason, the development and implementation of a comprehensive system for such automation requires changing the design technology itself as well as the training and education of a new generation of designers. For example, it is no secret that

the implementation of automated design techniques frequently runs up against psychological resistance from the designers.

An automated design system for computer hardware, which is incorporated in the sector level system (YeSAP) is being developed at the scientific research institute for computers. The basis for the developmental work is a systems approach which makes it possible to comprehensively resolve the problems of design, fabrication and servicing of the hardware as well as set up a through-going process of "top-down" design, combine manual, automated and computer aided design techniques into a single system, monitor all phases of the computer design, minimize the volume of stored information through setting up and managing data files based on data banks and create a set of special automated design system hardware to automate the major production and quality control processes for radio electronic equipment assemblies and units.

All of this provides for the creation of a single data base for the following: the informatin servicing of all of the tasks being carried out at an enterprise; the compilation, and frequently new compilation of a single set of documentation and computer data storage vehicles, which reflect the automated design results; assistance in improving the process of document turnover at enterprises as well as between them, which is related to the appearance of new forms of documents (punched tapes, magnetic tapes, etc.); enabling a direct transition from automated design of software using general purpose computers to the automation and the quality control of specialized automated hardware.

The major components of the automated design system are the central data bank (TsBD), the packages of applied programs (PPP), the interfaces and the control program. The nucleus of the automated design system is the central data bank. It consists of the control system and the data base. The latter contains the information needed for the design of the assemblies and equipment. The applied program packages perform logic functions as well as the functions of structural and production process design, and the design of hardware for the design processes. Modeling methods (system modeling, functional modeling, logic modeling) have become the most widespread in the logic design work.

The major problem here is the absence of special automated design technology which takes into account the capabilities of the automation hardware being used. For this reason, it is frequently necessary to dispense with the existing automation facilities and incorporate the engineer in the design process.

The logic design system includes the following subsystems: automation of the microprogramming, logic synthesis, and modeling at the level of the basic circuits. The automation subsystem for the microprogramming contains a translator for the microprograms from an ASSEMBLER type language as well as debugging (modeling) aids. Modeling aids at the levels of the structural circuits of the equipment and their transformations to logic schematics in the specified component base are contained in the logic synthesis subsystem. And finally, the modeling subsystem for the hardware at the level of the basic circuits makes it possible to compose a model from descriptions of the TEZ [standard electronic circuit modules], stored in the automated design system data bank, and to carry out the modeling process taking the average time parameters of the components into account.

The stage of the structural design of computer hardware has attracted the attention of numerous researchers as the most labor intensive step. However, it cannot be said that problem of design planning (KP) has been completely resolved. The known means of automating design planning do not as yet meet the existing requirements, especially in the large series production of products.

The large number of variables and the complex nature of the interrelationships elevate the problems of engineering design to the class of problems which permits solution at the present time only by means of heuristic methods. This is why the most acceptable approach to their solution at the present time is decomposition, which presupposes the breakdown of each task into a number of constituent tasks, the major ones of which are component assembling, layout and routing. This approach has been basically realized in the automated design system.

The programs for designer planning have been combined into subsystems intended for the design of various structures (standard structural modules, blocks of standard structural modules, frame, rack) using various types of assembly (printed circuit, thin conductor, hard wiring). The designer planning subsystems make it functionally possible to design standard electronic modules with printed circuit wiring, blocks of standard electronic circuit modules with printed circuit and thin conductor wiring as well as hard wired assembly with various means of fabricating the frames and racks and also make it possible to generate tests for checking the standard electronic modules.

In the various stages of computer aided design, the necessity arises repeatedly of checking the design process and editing the intermediate and final results. In this case, the user "accesses" the system by means of graphics interfaces. The generally adopted method of the tabular description of drawings using punched storage vehicles does not provide adequate operationally timeliness, and what is the most important, does not assure the reliability of the input and edited information. For this reason, a graphics coder has been developed and introduced in the Scientific Research Institute for Computers; this coder is based on an electromagnetic plotting board and a design display screen console. This unit makes it possible to encode the graphics information on the display screen and enter it in the designing computer in an operationally timely manner in an interactive mode. Where necessary, the data on the unit being designed is called up on the display screen and corrected there.

Some complex nonstandard printed circuit boards will still be designed manually for some time. The "Minsk-2004" and "Minsk-2009" periperals have been developed for encoding these circuit boards; these periperals make it possible to read graphics information from draft drawings.

When fabricating printed circuit boards, the most large-scale and labor intensive process is the production of the photographic templates. The number of them is figured in thousands of units, and the requisite fabrication precision reaches hundredths of a millimeter. A series of automated

peripherals have been developed to automate these processes. These are the "Minsk-2000", "Minsk-2004" and "Minsk-2005" photographic coordinatograhs, which provide for the automated preparation of photographic templates by means of a controlled light beam. A no less massive and labor intensive process is the drilling of holes in printed circuit boards. This problem has been successfully resolved by means of a specially developed printed circuit board drilling automat, the "Minsk-2003".

Quality control plays a special part in the fabrication of printed circuit boards by automated methods. It is accomplished by the "Minsk-2007" and "Minsk-2101" equipment developed by us. The former provides for visually inspecting the photographic templates and the second for completely checking the connections. The assembled and installed standard electronic modules undergo checking in accordance with a preset program in a special test stand. The automats which have been built also make it possible to have both punched tape and magnetic tape control of the processes. This is achieved through the specially designed "Minsk-2006" peripheral, which reads data out from a magnetic tape and feeds the control pulses out via a photosensitive reader channel.

More than 10 types of specialized equipment have been developed by the entire staff of the institute for the encoding and editing of graphics information, the fabrication of photographic templates, as well as drilling holes in printed circuit baords (PP) and performing a number of quality control operations.

We are happy with the overall results of the introduction of the automated design system. The timeframes for the design of individual assemblies and modules have been curtailed by a factor of two to three times and the process of developing new hardware has been considerably accelerated.

8225
CSO: 1863/181

PUBLICATIONS

TABLE OF CONTENTS OF UPRAVLYAYUSHCHIYE SISTEMY I MASHINY, MAY-JUN 83

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 pp 1-2, 128

[Table of contents of journal "Control Systems and Machines"]

[Text] Contents

9875
CSO: 1863/196

CONFERENCES

ALL-UNION SCHOOL OF YOUNG SCIENTISTS

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 p 127

[Article by O.L. Perevozchikova, "The All-Union School of Young Scientists", in the section "Symposia, Conferences, Meetings"]

[Text] The All-Union School of Young Scientists "Methods of Software Creation and Technical Realization of High-Performance Computers" was held Oct 25 to 29 1982 at Kiev. It was organized by the TsK VLKSM [Komsomol Central Committee], TsK LKSM Ukrainy [Ukraine Komsomol Central Committee], and USSR and UkSSR Academies of Sciences. The initiative for holding the school was assumed by the Institute of Cybernetics im. Glushkov of the UkSSR Academy of Sciences. It was devoted to the 60th anniversary of the formation of the USSR, 1500th anniversary of Kiev and 25-year anniversary of the Order of Lenin Institute of Cybernetics imeni Glushkov of the UkSSR Academy of Sciences (IK AN USSR).

The school indicates the growing interest in such cybernetics disciplines as information converter theory, system engineering, system programming and artificial intelligence. With the rapid growth in computer resources and faster rates of computer modernization (up to a change in generations), of primary importance is work to create system and application software and new architecture principles (including non-Neumann), structures and components of computers.

The school was a look into the future, since in 5-10 years the current young scientists will form the basic unit of cybernetics researchers, and the level of their ability will determine further improvements in computers and problem solving tools using the computer for the period to 1990.

About 200 young scientists and specialists from 11 union republics participated.

Deputy director of the IK AN USSR, doctor of technical sciences V.P. Derkach and secretary of the Kiev gorkom LKSMU V.P. Babak opened the school.

Leading academicians delivered review papers on achievements in cybernetics at home and abroad at the plenary meetings: academician of the AN USSR [UkSSR Academy of Sciences] V.L. Rvachev, corresponding member of the AN USSR Ye.L. Yushchenko, doctorsof physical-mathematical sciences Yu.V. Kapitonova, A.A. Letichevskiy and I.N. Molchanov, and doctors of technical sciences I.D. Voytovich, V.P. Derkach, V.M. Yegipko and S.B. Pogrebinskiy.

With great interest, the school's participants learned of new developments at the IK AN USSR such as the macropipeline computer complex, computer family with hardware support of the software (including the SM-1410 two-processor complex with the ANALITIK-79 input language), technology for manufacturing cryoelectronic elements based on the Josephson effect, microprocessor tools, and scientific research automation systems.

Documentary films were also shown at the plenary meetings on developments of the IK AN USSR on automating scientific research and "Academician V.M. Glushkov: Cybernetics Specialist".

The works of young scientists and specialists from several of the country's leading enterprises and institutions were actively discussed at the topic meetings. These were held in three sections: computer component base and architecture; artificial intelligence and the use of high-performance computers; and high-performance computer software.

The largest number of papers and reports were presented at the third section, reflecting the growing work on software in designing and creating computer tools. Of greatest interest in this section were papers examining methods of data processing, aspects of machine and system design, organization of computations, problem-oriented complexes, methods of implementing language processes, and the experience of developing the "El'brus" multicomputer complex's software.

During the school, trips were organized to the computer center of the IK AN USSR, and a specially organized exhibit of the best works by young scientists of the Institute.

In the opinion of the participants, the young scientists school was held at a high scientific level and successfully performed its basic function: to acquaint them with the experience and achievements of researchers at home and abroad, establish scientific contacts, and help raise the skills and performance of young scientists in methods of creating software and technical realization of high-performance computers.

It is recommended that the next school be held in 1984, based on the country's leading scientific teams.

9875
CSO: 1863/196

PERSONALITIES

60TH BIRTHDAY OF V.P. DERKACH

Kiev UPRAVLYAYUSHCHIYE SISTEMY I MASHINY in Russian No 3, May-Jun 83 p 93

[Announcement of the 60th birthday of V.P. Derkach by the editorial board]

[Text] On April 13, 1983, employees of the IK AN USSR [Institute of Cybernetics, UkSSR Academy of Sciences] and representatives of many scientific and academic institutions and organizations of the republic warmly congratulated deputy director of the IK AN USSR, laureate of the USSR State Prize, Distinguished Scientist of the Ukrainian SSR, Doctor of Technical Sciences Vitaliy Pavlovich Derkach on his 60th birthday, and wished him good health and new creative successes in developing the component base of new generations of computers and in training scientific personnel.

The editorial board of this journal joins in these congratulations and good wishes.

9875
CSO: 1863/196

- END -